GAME THEORY AND U-BOATS IN THE BAY OF BISCAY

THESIS

Joseph C. Price, Captain, USAF

AFIT/GOR/ENS/03-18

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/03-18

GAME THEORY AND U-BOATS IN THE BAY OF BISCAY

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Joseph C. Price, BSAE

Captain, USAF

March 2003

AFIT/GOR/ENS/03-18

GAME THEORY AND U-BOATS IN THE BAY OF BISCAY

Joseph C. Price, BSAE
Captain, USAF

Approved:

____________________________________ __________________
Dr. Raymond R. Hill (Chairman) date

____________________________________ __________________
Dr. J. O. Miller (Member) date

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Ray Hill, for all of his patience, advice, encouragement and understanding, without all of which I would not have succeeded. I'd also like to thank Major Paul McAree for his instruction in game theory, and Dr. J.O. Miller, for taking on reader's duties at the last moment. I must also thank Major Lance Champagne and Capt Greg Carl, my research buddies, for all their guidance and patience, especially these last few months.

I would like to thank all of my classmates and friends here for helping an "engineer" make it through this program. Especially the following: Kris and Yager for the bloody climbing trips, teaching me how to properly wield a hatchet and build a fire; Kris for his RUSCO card; Yager for waking me up for Maui and leaving ice in my fridge; Justin for a great parking spot, blarney stones, and captain morgan; Scott for the rent payments and drinking Yager's leftovers; Don for his faith in the 28th and all the workouts; Mo for finding my PDA and ruining the lab with his anger; and Susie for the platypus. Many thanks go to the Hoffman's, the Pruitt's, the McGee's and the Hill's for taking in a single guy and looking out for him. And to all my classmates here, with whom I have laughed, ate, drank, and played euchre with over the last 18 months: you kept me mostly sane.

To my faraway friends: Rick, Jason, Dave & Shannon, Kerry & Erin, Aaron, Bill, Matt, Jamie, Paul, Tara, Wendy, Mel, Scotty, Troy and Jeanette; thanks for the support and the faith. You all helped me get through this more than you know. And finally, to my parents and my younger brothers, thanks for your love, understanding and support during this busy time.

Joseph C. Price

Table of Contents

List of Figures

List of Tables

AFIT/GOR/ENS/03-18

Abstract

Analysis of a large combat campaign using game theory is difficult due to non-linearities and other soft factors which exist in a complex system. However, game theory can give decision makers insight into strategies and outcomes that can be utilized to maximize one's objective. Agent-based simulation provides the means to model complex systems with non-linearities, by allowing for interactions among independent "agents." This thesis investigates game-theoretic strategies in agent-based simulation, modeled after the Allied search for U-boats in the Bay of Biscay during World War II (WWII). It also looks into the effects of adaptation on strategies by comparison to fixed-strategy results.

GAME THEORY AND U-BOATS IN THE

BAY OF BISCAY

I. Introduction

1.1. General Issue

Models have been consistently used by the military for a wide variety of applications. Such applications range from acquisitions and force structuring, to capability analyses and combat. Combat models are of primary interest to the military; they are used to analyze and investigate combat development, operations and training. However, no model is perfect and accurately reflects the "real world"; the greatest limitation of combat models is that they do not reflect complex or chaotic behaviors such as heroism, leadership, or morale.

Recent advances in computing capability, the development of new computer language architectures, and a shift in thinking paradigms have resulted in the appearance of agent-based technology. Agent-based architectures promote the modeling of complex systems, by enabling non-linearities and interactions to occur between independent, interacting "agents". Agent-based models can be applied to combat scenarios, and allow for complex behaviors to emerge, which were not possible to capture from traditional models. Agent-based models are still relatively new, and must be rigorously examined to realize and understand their potential.

1.2. Background

Models are physical, mathematical or logical representations of systems, possibly simplified in some way, to gain insight into how the system behaves, and even predict future behavior (Law, 2000:2). Traditional models generally fall into one of the following categories: 1) physical models that deal with real-world objects; 2) analytical models, which are defined in terms of exact relationships and quantities and produce an exact result; 3) simulations, which are usually models of complex systems allowing the analysis of different input and output parameters (Law, 2000:4-5).

Combat models are models, such as simulations, that reflect elements of military operations for investigative or management purposes. Combat models are used by the military to aid in decision making and can affect force structure, budget allocation, acquisition of new systems, force deployment, tactics and strategy. Such models generally fall into one of three categories: combat development, operations, and training. Combat development models are used to facilitate research and development in new technologies and new weapons systems. Operations models provide understanding of the real world, through channels such as war planning, policy analysis, and historical analysis. Finally, training models are employed to prepare military personnel and units for combat.

The Department of Defense also classifies combat models into a hierarchy as shown in Figure 1. Models are classified at the various levels depending on their scope of contribution. The levels are: specialty, engagement, mission and campaign. As one progresses up through the levels, the models become increasingly aggregate. Models at the specialty level are highly detailed, and include engineering models, individual combat systems or subsystems. Engagement level models reflect the interactions between combat systems, such as one-on-one aircraft battles, and can deliver high levels of detail on the systems of interest.

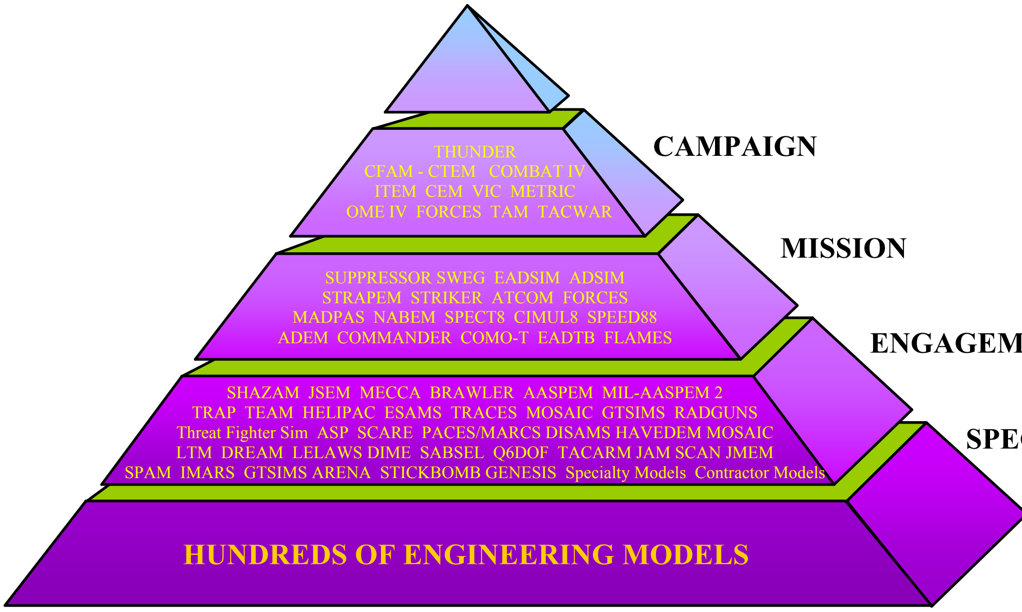


Figure 1. Hierarchy of Models (Miller, 2002)

Mission level models can simulate traditional combat (land, sea, and air) in a region or theater of operations for a short time period. Most mission level model conflicts range from few-on-few to many versus many, and give insight into tactics, strategies, and other mission areas of interest such as command and control. Finally, campaign level models simulate traditional combat in the region or theater over long periods of time. Often, campaign models incorporate analyses of logistics and support systems, attrition, strategies and additional factors which may affect the ability to sustain wartime operations in a regional conflict.

There are drawbacks to using models in general, especially combat models and simulations. One drawback is the inherent limitations of the model itself. Traditional combat models are built around mathematical relationships, quantities, and logic. Where they falter is in the capability to quantify factors that cannot be explicitly expressed, concepts such as leadership, courage, morale, and other human traits. These models also display very little adaptability in two aspects. First, the objects in the model cannot react and change to conditions beyond specifically encoded rules. Second, the applicability of these models and their ability to function validly for alternate scenarios are restricted:

> A danger inherent in detailed, complicated computer simulations of combat is that they reflect current doctrine; their ability to intimate radically new tactics is limited (Hughes, 1984).

As the military has evolved away from the traditional type of warfare into operations other than war, information warfare and space warfare, new types of models are desperately needed.

A second drawback involves the interpretation of results. Many people expect simulations to give *the* correct answer, or *the* way to accomplish various tasks. Instead, decision makers must realize that models provide insight into how a system operates and the importance of various factors which are represented in the system, not predictions of an outcome. This assumption about models and simulations has the potential to be very harmful, especially when the system contains human factors.

The recent development of agent-based models has provided means to circumvent the aforementioned model limitations. Agent-based models have demonstrated the capability to model non-linearities, interaction and chaotic elements, develop emergent and complex behaviors, and allow for adaptation within the system. These characteristics will benefit combat models and enable research into areas (with human behavioral and cognitive characteristics) not possible before. The potential for these agent-based models, especially combat models, needs continued exploration, along with compatibility with applications of other techniques such as search theory and game theory.

Game theory is the study of conflict, through which opposing players, their strategies and choices, and the possible conflict outcomes are examined. Game theory has been used to study a variety of topics including economics, business, and politics. Additionally, game theory is a useful tool for analyzing combat and the methods by which opposing sides operate. It is thus reasonable to investigate the combined impacts of game theory and agent-based modeling.

1.3. Research Objective

The objective of this research is to explore the effects of game theoretic strategies in a campaign model as embodied in an agent-based model. German U-boat operations in the Bay of Biscay during World War II, and the subsequent Allied search efforts to counter these operations, provide the scenario for the agent-based simulation. The model is designed using an object-oriented computer language, and guided by historical data. The results of the simulations are analyzed to determine the effects of different strategic scenarios on the effectiveness of hunting U-boats in the Bay of Biscay, and relating these results to historical perspectives.

1.4. Scope of Research

This research examines the daytime versus nighttime searching strategies of the Allied air forces and the surfacing strategies of the U-boats represented in the model. Three different scenarios, ranging from fixed to adapting strategies, are constructed to compare results and validate game theoretically-based logical conclusions. This research also investigates the development of resultant data landscapes, which are examined for potential game theoretic critical points.

1.5. Overview of Thesis

Chapter 2 of this document will provide a history of U-boat operations in the Bay of Biscay during World War II, and the antisubmarine search campaign conducted by the Allied air forces. This chapter will also summarize a few previous analyses of this campaign. Chapter 3 presents basic definitions on the topics of agent models, game theory, and search games, and reviews some of the published literature. Chapter 4 is a description of the baseline agent model used for this project, and the relevant assumptions underlying the model. Chapter 5 looks at the enhancements incorporated into the baseline model and sets up the methodology for three scenarios investigating game theoretic strategies. Chapter 6 discusses the results of the simulations and presents an analysis of these results, and Chapter 7 outlines model and analytical limitations and potential future research in this topic area.

II. Background

2.1. Bay of Biscay Scenario

2.1.1. Historical Overview

During World War II (WWII), the Germans were the first to effectively use the submarine against non-military targets, specifically against the logistical forces supporting the Allied war effort. U-boats (from the German word for submarine, *Unterseeboote*) were, in fact, used primarily to sink Allied merchant ships crossing the Atlantic Ocean to re-supply the Allied forces in Europe. For a period of time in 1943 and 1944, the U-boat effort was so effective and devastating to the Allies that later Winston Churchill wrote that "the only thing that ever really frightened me during the war was the U-boat peril" (Churchill, 1949).

From 1941 through 1944, U-boats operated out of captured ports on the western coast of France. From these ports, the U-boats transited the Bay of Biscay to the Atlantic where they hunted convoy targets. The Bay of Biscay is bordered on the east by France, to the south by Spain and Portugal, and in the north by Great Britain and Ireland (Figure 2). While departing from and returning to their ports in France, U-boats spent a significant amount of travel time in the Bay, as it provided the only access route to the Atlantic.

It was in these waters that the Allies decided to concentrate their search effort in an offensive endeavor to counter the U-boat threat. The Bay, with a few exceptions, was the only feasible area to conduct U-boat hunting operations. The open waters of the Atlantic were simply too vast providing ample area for U-boats to navigate and hide. The German occupied ports in France were heavily defended and hardened against bomber attacks. Additionally, German fighter aircraft patrolled the skies over and around the ports, deterring direct attacks against the ports or attacks against U-boats in the coastal waters near France.

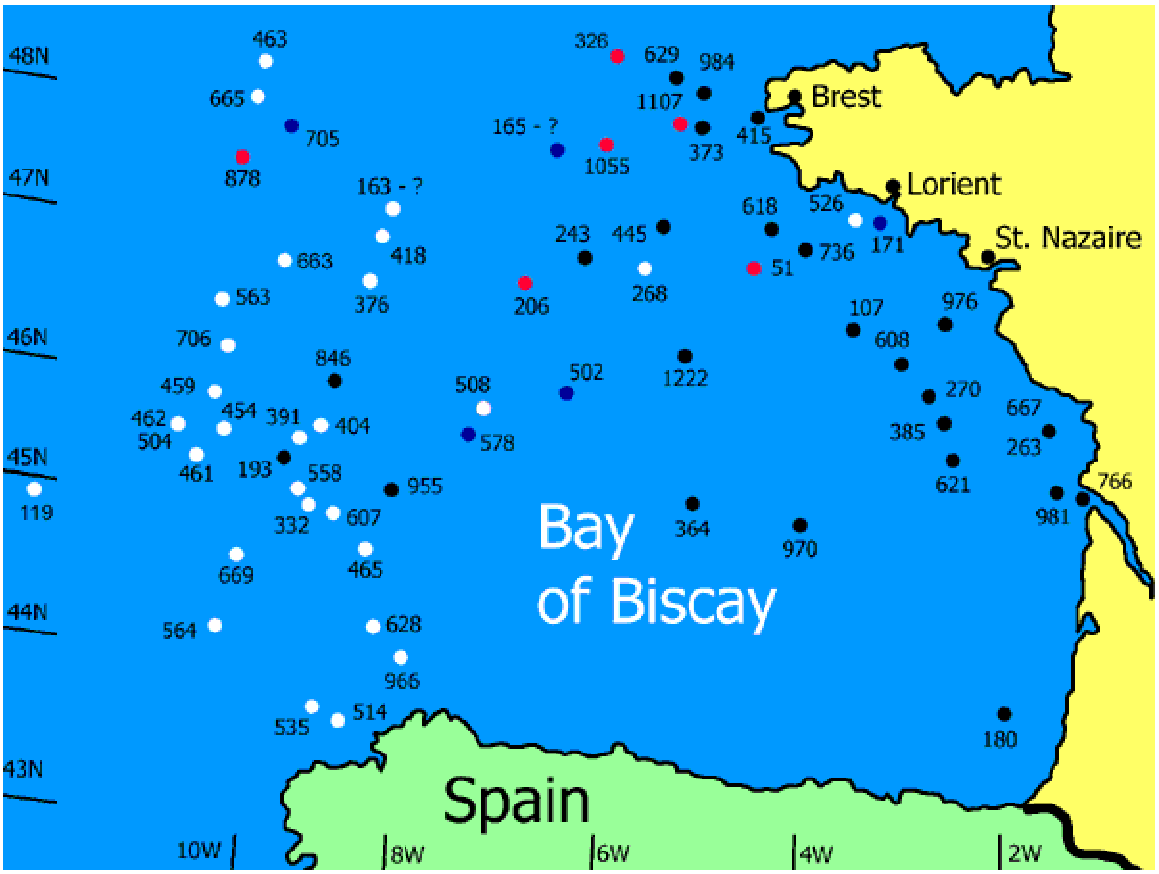


Figure 2. Map of the Bay of Biscay with Locations of U-Boat Sinkings (http://uboat.net/maps/biscay.htm)

There were two exceptions to this policy: near the merchant convoys and at U-boat re-supply points in the Atlantic. Escorting merchant convoys could only be accomplished close to Britain, within the range of the Allied aircraft. This was not as effective as searching the Bay because U-boat captains generally engaged in attacking the convoy ships in the Atlantic before this point. The decryption of the German radio code, Enigma, by Allied intelligence allowed aircraft to locate and patrol several U-boat re-supply points in the Atlantic. A small portion of open-sea U-boat missions would meet with specially designed tanker U-boats to take on fuel and supplies thus extending their time in the Atlantic.

Although it was the primary feasible search area for the Allies, the Bay of Biscay still contained roughly 130,000 square miles of potential search area. The advantages the Allies gained by flying patrols here, however, made the effort worthwhile. U-boats, as mentioned previously, were forced to transit the Bay to and from the ports in France. Thus, the U-boat density in the Bay was higher than in any other potential search area. Also, submersible technology at the time did not enable submarines to stay underwater indefinitely. U-boat engines ran on two sources of power, diesel engines and batteries. Batteries only enabled the U-boat to travel approximately 100 nautical miles underwater before it was forced to surface for nearly three hours to recharge the batteries while operating under diesel power. As a consequence, U-boats were forced to surface at some point during their transit of the Bay. A surfaced U-boat was much easier to locate than a submerged U-boat.

Between 1941 and 1944, Allied air forces vigorously patrolled the Bay. Early on, the finding and "killing" of a U-boat was an infrequent event; however, as the development and implementation of new technologies such as radar progressed throughout the war, the Allied search forces became more adept at finding and sinking U-boats. To reduce passages through the Bay, the Germans employed the use of tanker U-boats, modified to carry fuel and supplies to U-boats already in the Atlantic. These tanker U-boats could re-supply up to ten U-boats, enabling each to spend another 30 days at sea. Without refueling, a U-boat could spend 30 days at sea; refueling thus effectively doubled their time (multiple refuelings were possible) in the Atlantic. However, tanker U-boats were limited in number, and Allied air forces, recognizing them as a potent aid to the German effort, hunted tanker U-boats relentlessly.

Returning to the ports in France, besides forcing U-boats to transit a patrolled Bay, was also dangerous as there was the risk that the U-boat may never put out to sea again. The German U-boat repair facilities were plagued with a lack of maintenance technicians and supplies, and efforts made by German leadership to correct these shortcomings ultimately failed. Maintenance backlog grew throughout the war, and a U-boat returning from sea faced an increasing chance of spending the rest of the war in dry-dock waiting for repairs.

2.1.2. Campaign Issues

So what is the importance of the Bay of Biscay campaign? This conflict between the Allied search aircraft and the avoiding German U-boats is rich with strategic and tactical decisions on both sides, driven predominantly by the state of technology and its progression. Additionally, the field of *operations research* was born out of the Allied attempt at modeling the conflict and optimizing their search strategies.

This campaign has been described as a technology "see-saw" or "tug-of-war" (McCue, 1990:8). Until the Germans began outfitting the U-boats with snorkels, allowing them to stay underwater indefinitely, the technological focus was on the Allied use of radar for searching and the radar countermeasures employed by the U-boats.

The result of technology battles forced larger tactical and strategic issues on both sides. Since U-boats could not transit the entire Bay submerged, their issue was when, and for how long, to surface. U-boats traveled faster on the surface than underwater (average of 10 knots versus average of 2.5 knots), and less time spent in the Bay meant more time to operate in the Atlantic. Increased surface exposure, however, increased the risk of being detected and destroyed by Allied aircraft. Traveling underwater reduced their vulnerability, but greatly increased the transit time. While surfacing at night was usually safer, a strategy of only surfacing at night would allow Allied search efforts to concentrate solely at night. Finally, the use of countermeasures in the form of search receivers could give U-boats advance warning of approaching aircraft. The downside was that a side effect from the use of these primitive search receivers was the broadcast of radio waves that Allied aircraft could detect and track.

The Allied search forces were faced with similar issues: the time of day to search, where to search, and how often to search a given area with a limited set of resources. There was a vast difference in the effectiveness of day searching versus night searching. At one point in the campaign, it was calculated that the probability of killing a sighted U-boat during the day was approximately 40% while at night it was merely 11% (Waddington, 1973:201). If Allied search effort were restricted to day searches, when Allied aircraft effectiveness was higher, the U-boats could capitalize by surfacing only at night. Such were the game theoretic issues facing Allied decision makers during the war, and in particular this aspect of the war.

2.2. Previous Analyses

The first analysis of this Bay of Biscay campaign was written in 1946 shortly after WWII, but not cleared for publication until 1973. In *O.R. in World War 2*, author C.H.

Waddington details the efforts of the Operational Research Section (O.R.S.) of the Royal Air Force Coastal Command in countering the U-boat threat. His personal involvement in that effort provides firsthand insight of how various techniques were applied to the problem at hand to promote effective management decisions – hence *operations research* was born. The book describes how the O.R.S. engaged a wide variety of topics, from radar to navigation issues to weather, altitude and attack methods, with a goal of demonstrating that "scientific methods of analysis might give useful assistance to effective executive action" (Waddington, 1973:xii).

McCue (1990), re-examines the analyses that were accomplished during WWII, and in some cases completing them with modern techniques. He addresses a wide range of issues from the technology "see-saw", seasonal impact, and the concept of a balanced search effort. McCue's work shows that "one can quantify and analyze the campaign against the U-boats not merely by applying..."index numbers" and "coefficients based on sound military judgment"…but through mathematical reasoning systematically applied to knowable physical quantities" (McCue, 1990:2).

2.3. So Why This Scenario?

The Bay of Biscay campaign is useful as an agent-based model for several reasons. First, the amount of information available on the subject is immense. Besides the first-hand account analysis of Waddington and the later analysis of McCue, other sources, including Grand Admiral of Submarines Karl Dönitz's *War Diary*, are accessible. Dönitz's source is valuable as it gives insight into German U-boat strategies and tactics utilized in the Bay, as well as numbers for comparison. This availability of information lends itself to the development of a detailed model, and is also useful for the comparison of model results with historical results. Second, the technological developments, and the tactical and strategic decisions necessitated by this campaign raise a lot of "what if?" questions. These questions can be investigated by such a model with tradeoff analysis techniques. And finally, the model of a historical campaign like this, which is essentially a type of predator/prey model, will easily transition to a variety of present-day scenarios for investigation. Such scenarios include immigration, drug-running, smuggling, and terrorism, as well as some more traditional military operations.

III. Literature Review

There are three key concepts incorporated into this project. Each topic individually has a vast body of literature and discussion. This chapter thus summarizes the fundamentals of each topic, their development and current applications. Since the simulation program developed for this project is an agent-based simulation, this review begins with an assessment of *agent-based modeling*. The objectives of the project center on the application of *game theory* to the agent-based model, therefore after a brief discussion on game theory, focus shifts to *search games,* game theory with search applications.

3.1. Agent-Based Modeling

Classical analysis techniques and models used in most simulations are largely grounded in linear systems. Naturally, when it comes to the analysis of non-linear or complex systems, these linear systems are inadequate. Non-linear systems are those in which certain characteristics of the system cannot be linearly decomposed into the characteristics of the system's components. In other words, the sum of the whole is greater than the parts. A fairly new method of modeling, agent-based modeling, utilizes small building blocks, known as agents, to represent objects within the system. Allowing these agents to interact with each other, and with their environment, emulates complex systems.

3.1.1. Definitions

Agents can be defined as entities which are "distinguishable from [their] environment…possess some kind of identity…have some autonomy of action, that they can engage in tasks in an environment without direct external control" (Rocha, 1999:3). Agents are governed by a set of rules to function, on what they can "sense" and how they can act. Agents must also act rationally, based upon what they can perceive (Russell and Norvig, 1995). Agents are used to represent a variety of objects in different systems and study areas. However, most experts agree that agents must possess the basic qualities mentioned above. Ferber (1999) gives the following well-developed definitions for the qualities of an agent and multi-agent systems. An autonomous agent is a physical or virtual entity

- that is capable of acting in an environment;
- that can communicate directly with other agents;
- that is driven by a set of tendencies (has autonomy);
- which possesses resources of its own;
- that is capable of perceiving its environment (although limited);
- that has only a partial representation of this environment;
- which possesses skills and can offer services;
- that may be able to reproduce itself; and
- whose behavior tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representations and the communications it receives.

A multi-agent system has the following elements:

- an environment;
- a set of objects that can be perceived, created, destroyed and modified by the agents;
- an assembly of agents (the active entities in the system);
- an assembly of relations linking the agents;
- an assembly of operations enabling agents to perceive, produce, consume, transform, and manipulate objects; and

- operators whose task is to represent the application and reaction to these operations.

Agents are commonly classified into two types: reactive (or dynamically coherent) and adaptive (dynamically incoherent) (Rocha, 1999:2-3). Reactive agents are so named because they can only "react" – their next action (or state) is based on their current state and the state of their environment. They have some degree of autonomy, but it is closely coupled with the environment. Adaptive agents, however, possess some form of *memory* which allows them to make decisions based not only on their current state or the environment, but also on information that is stored within their memory. This information can be merged using some type of decision procedure, allowing a decision capability that may produce different behaviors in agents. This ability to possess alternate behaviors enables the agents to adapt over time beyond their initial state and may result in unexpected system behavior.

Multi-agent systems are systems which contain more than one agent. Multi-agent systems usually possess the following basic characteristics: agent goals, an environment, interaction of some type, and observed behavior. Agents each have individual goals, some of which may be in conflict with the goals of other agents. Not all of the agents need to be of the same type. Most real world systems have more than one type of entity, and so are appropriately modeled using multiple types of agents. The agents, as a group, may also have a group goal, which may not align with their individual goals.

The environment in which the agents are contained must allow the agents to interact with it, and allow the agents to interact among themselves. The environment is often user-defined, its resolution being set based upon the objectives of the model. For example, an environment may be as simple as a rectangular grid in two-dimensional space, or as complex as a 3D terrain map for a battlefield complete with weather, terrain effects, and electronic warfare capabilities. The environment can also be adaptive, allowing the agents to act upon it and alter its current state.

The most important function of the environment is to allow interaction. Interaction between the agents and the environment is facilitated by simple rules and bounds, which are imposed upon the agents and, if required, by the agents on the environment. The environment also facilitates interaction between the agents, by allowing some form of communication or knowledge to pass from one agent to another. Interaction is the most important aspect of multi-agent systems. It is this interaction property that enables the agents to react and adapt, and develop agent behaviors. Interaction also allows the system to develop behaviors that are not developed by the individual agents. This *emergent behavior* is the subject of study in what is known as the *Science of Complexity*, which examines the behaviors of complex systems. This is useful in modeling such complex systems as societies, economies, or businesses (Jilson and Mert, 2001:7).

3.1.2. Applications of Agent-Based Modeling

Various fields of research and study have applied the concepts of agent-based modeling to gain insight into the systems of interest. Computer science has used agents to research aspects of *artificial intelligence* and robotics, as well as computer and internet networks. Russell and Norvig (1995) discuss in depth on how agents and agent systems are related to artificial intelligence. Sycara (1998) also reviews several computer-related applications of multi-agent systems. Batty and Jiang (1999) look into how multi-agent simulation has been used in geographic information systems. Rocha (1999) details a number of agent-based models and their contributions to the field.

An advancement often tied to agent-based modeling is the development of *data farming*. This technique, outlined in Brandstein and Horne (1998), is the result of increased computing power and the flexibility of agent-based modeling. It is the ability to conduct bottom-up analysis of a system by altering characteristics of that system at the agent level. Each run of the system provides a point in "the landscape of possible outcomes" for a given measure of effect (Brandstein and Horne, 1998:95). Considering all parameter settings yields this *landscape of outcomes* used to gain insight into the system.

It is no coincidence that data farming was developed as part of *Project Albert*, at the US Marine Corps Combat Development Command (Horne, 2001). Traditionally, combat has been modeled as various linear systems, or alternatively by the Lanchester Equations. The Lanchester Equations are a set of coupled ordinary differential equations that model warfare attrition (Ilachinski, 2000). However, combat and warfare are anything but simple and deterministic; they are chaotic, complex systems characterized with intangibles such as the Clausewitzian fog and friction of war, courage, human leadership and such. Agent-based models can be applied to combat not to give a prediction or an outcome, but to enhance understanding of combat. *Project Albert*, for example, seeks to "develop and apply a series of new models…to explore, and seek robust answers to, questions relevant to Marine Corps organization, equipment, tactics, and doctrine" (Horne, 2001:2).

One of the first examples of agent-based modeling applied to combat is ISAAC, or Irreducible Semi-Autonomous Adaptive Combat. Sponsored by the Marine Corps, and developed at the Center for Naval Analysis, ISAAC is a computer simulation designed as a "conceptual playground" of agent-based combat modeling, and not a full combat model (Ilachinski, 2000). The designers used a bottom-up approach versus the traditional top-down design, and each agent in ISAAC represents a primitive combat unit, governed by different "personalities" and "meta-rules". Ilachinski (2000) further discusses interesting complex behavior to emerge from simple primitive actions. EINSTein, a follow-on to ISAAC, extends the model as "an interactive tool box for the general exploration of combat as a complex adaptive system" (Ilachinski, 2000:41).

3.2. Game Theory

3.2.1. Background

Game theory is the science of conflicting interest (Luce and Raiffa, 1957). The mathematical approach of game theory originated in the early 1900's from mathematician and economist John von Neumann, who published papers on game theory in 1928 and 1937. The main work of impact was the book by von Neumann and Morgenstern, *Theory of Games and Economic Behavior* (1944).

A conflict of interest exists when two or more "individuals" have a situation where decisions must be made. The situation may result in several possible outcomes, dependent on the decisions made by the individuals. These individuals all have their own preferences on which outcome they desire, or value higher, and their preferences are not in agreement. Game theory seeks to analyze the components of these conflicts, and describe the choices of each individual and the possible resulting outcomes.

Game theory outlines a few characteristics common to all *games*. First, the game has two or more rational players, some of which are competing against each other. Second, there is a payoff, or utility, that each player desires to maximize. This utility may be represented by a number of things: for instance money, time, or effort. To accomplish the goal of payoff maximization, each player must make certain decisions during the game, sometimes without knowing what decisions the other players will make. The propensity, or probability, to make a certain decision is a player's strategy. Through analysis of the game, game theory attempts to determine which strategies will allow the player to maximize their payoff, usually at the expense of the other players.

3.2.2. Definitions

The assumption of "*rational*" players is the basis for conclusions in game theory. Rationality possesses many meanings, and differing levels depending on the environment in which the decision or choice of action is made (Rapoport, 1996:55). Although there is no universal definition of rationality, Luce and Raiffa define it as the following:

> Of two alternatives which give rise to outcomes, a player will choose the one which yields the more preferred outcome, or, more precisely, in terms of the utility function he will attempt to maximize expected utility (Luce and Raiffa, 1957:50).

Closely linked to the assumption of rationality is the quality of information. Some games are "*games of perfect information*," where at any point in the game each player knows the moves or actions, their own and their opponent's, which have brought them to the current position. On the other hand, games without perfect information may conceal the available choices or the actions taken by one's opponent (Rapoport, 1966:62). For example, many card games such as poker or euchre are considered games without perfect information due to the concealment of one's cards from the other players.

Many games are characterized by two players who are opposed in preferences of outcomes; for example, if player 1 prefers outcome *x* to outcome *y*, then player 2 prefers outcome *y* to outcome *x* (Luce and Raiffa, 1957:59). These players are called *strict adversaries*, and if all of their outcomes oppose one another, the game is called a *strictly competitive* game. Luce and Raiffa remark that if a strictly competitive game can be represented such that the sum of the players' payoffs sum to zero, the game is known as a *zero-sum game* (Luce and Raiffa, 1957:64).

A player's *strategy* is the representation of the choices they will make throughout the game, and their set of strategies enumerates all the ways a game can be played to reach an outcome. For even the simplest of games, this set of "*pure*" strategies can be very large. A *mixed* strategy occurs when a player randomly chooses between two or more pure strategies with a given probability for each strategy. The objective of a mixed strategy is "to keep the opponent guessing about what one will do" (Rapoport, 1966:69).

In some games, there may exist what is called a "*saddle point*" or an "*equilibrium point*." This point occurs where both rational players have chosen strategies such that it does not benefit them to change their strategy if their opponent does not change their strategy (Luce and Raiffa, 1957:62). For example, consider a two-player zero-sum game. Player 1 wishes to maximize his payoff, and so player 2 wishes to maximize his own playoff, or because it is a zero-sum game, he desires to minimize player 1's payoff. Player 1 is a maximizing player, while player 2 is a minimizing player. If an equilibrium point exists, it is the point where the player 1 has chosen a strategy that maximizes the minimum payoffs from each of his strategies. Simultaneously, player 2 has chosen a strategy that minimizes the maximum payoffs from each of his strategies to player 1. Hence, this point is sometimes called a "minimax" point, but more often it is referred to as an equilibrium, or saddle, point. Equilibrium points are not necessarily unique for a game, and may exist for games with pure strategies or games with mixed strategies.

The applications of game theory are wide-ranging. Common applications include economic and business situations, politics, behavioral sciences, and combat. Although many assumptions have to be made between the transition from theory to real world, game theory can generate valuable insight into the nature of the conflict and the decisions facing the players.

> [Game theory] prescribes for given assumptions courses of action for the attainment of outcomes having certain formal "optimum" properties. These properties may or may not be deemed pertinent in any given real world conflict of interest. If they are, the theory prescribes the choices which *must* be made to get that optimum (Luce and Raiffa, 1957:63).

3.3. Search Games

3.3.1. Definition

Search games are situations where a search problem can be formulated as a game theory problem (Benkoski and others, 1991:479). The conflict involves a searcher and a hider, or evader, who does not want to be found. The hider may be stationary or mobile. The payoff could be whether the hider is found or not, or how long it takes to find the hider. Benkoski et al. (1991) detail a good number of search game references in their survey of search theory literature.

3.3.2. Relevant Applications

The problem of searching for an evading target is addressed in Dobbie (1975). The problem space is defined as a two-cell problem, where a single evader tries to avoid detection by moving away from the searcher when he senses the searcher's presence. Dobbie finds the strategy for the searcher's effort that maximizes the probability of detection in a given amount of time. Stewart (1981) extends Dobbie's formulation with two special cases. First, the evader has a goal or objective to complete, and when this occurs, continuation of the search is no longer productive. Second, the searcher is subject to a constraint on his set of resources, prohibiting him from searching during every time period of the game.

Washburn (1979) considers a similar problem, except the target is moving in a discrete time and space. During each time step, the searcher tries to detect the evader, and to maximize the change of immediate detection (a *myopic* strategy). Washburn gives a necessary condition for optimality in this case.

Baston and Bostock (1989) approach a one-dimensional helicopter versus submarine game, modeled as a two-person zero-sum game. Both players are forced to move along a straight line, and neither player can see the other. The helicopter has a given number of bombs with which to attack the submarine, and the payoff is whether or not the submarine is destroyed. Baston and Bostock solve the game when only one bomb is available, and extend it to multiple bombs given certain constraints.

Eagle and Washburn (1991) address two-person zero-sum search games where play continues for some period of time without either player receiving feedback. Each player moves from a preset plan. The authors present two methods to solve the game: a method of fictitious play and a method of linear programming.

IV. Model Description

4.1. Baseline Description and Design

The baseline program was primarily authored by Major Lance Champagne, USAF, with contributions by Captain R. Greg Carl, USAF, and the author. The Bay of Biscay simulation was coded in JAVA, which as an object-oriented computer programming language, naturally lends itself to the creation of multi-agent simulations. The simulations were run on 2-GHz Pentium 4 PC's running a Windows 2000 operating system. Program design data was researched and utilized in the following order of importance: 1) historical fact as found directly from sources credited to Allied and German participants; 2) published studies directly related to the offensive search in the bay; 3) data derived from raw numbers in one or more of the preceding sources; and 4) good judgment (operational expertise) when the three previous sources fail or contradict one another (Champagne and Hill, 2003).

4.2. Relevant Assumptions

As with any model, there are certain key assumptions that must be stated up-front. Simulated time is updated in 2-minute increments. "Daytime" in the model is defined as the period between nautical sunrise and nautical sunset (nautical sunrise and sunset occur when the sun is 12° below the horizon). Each aircraft and U-boat represented in the simulation is an independent agent, having "fleet-level" or "squadron-level" properties or knowledge. This means that every aircraft has the same airspeed, detection capabilities, and such, but each acts independently of any other agent. Both aircraft and U-boat agents have detection sensors which correspond to the inverse cube law.

Aircraft search a 200 x 350 square nautical-mile (NM) area which is subdivided into twenty-eight 50 x 50 square NM non-overlapping sections. At least one aircraft per day is assigned to search each section. There are a maximum of 40 Allied aircraft operating out of bases in southern England. Each aircraft flies to their assigned search sections, avoiding the coast of France and within range of enemy fighter aircraft. Based on Allied aircraft assumptions we do not model enemy fighter aircraft. Aircraft takeoff times are randomly generated per simulated day (24 hours), with at least 12 hours between the previous day's landing time and the current day's takeoff time. Any maintenance cancellations occur before takeoff, and delay the aircraft for one day. Weather cancellations affect the entire flying day and cancel all sorties for that day.

An aircraft travels at 120 nautical miles per hour (knots), and will fly until it has used 70% of its fuel capacity, or until it has expended its munitions attacking a U-boat. Aircraft can only detect U-boats when the U-boats are surfaced, and will attack any detected U-boat if it is within range. Aircraft expend their entire munitions load in a single attack, and immediately return to base. Aircraft fly solo, and do not communicate with other aircraft. Aircraft use a crossover barrier patrol pattern when searching an assigned section (Figure 3). Attrition of aircraft due to accidents or active U-boat defenses is zero.

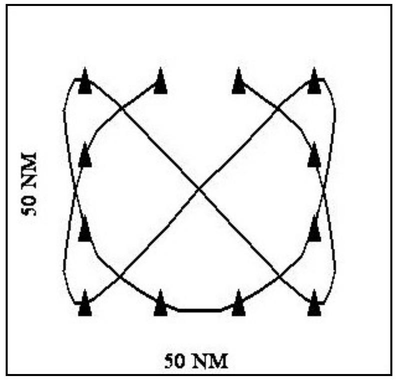


Figure 3. Crossover Barrier Pattern

The simulation starts with all 70 U-boats uniformly distributed throughout the Bay, one-half heading towards the North Atlantic, the other half heading to their home port. U-boats are assigned evenly to one of five ports in France. U-boats entering a port remain there for 25 to 40 days of maintenance. U-boats travel at 10 knots on the surface and at 2.5 knots while submerged. For every 100 NM traveled while submerged, U-boats must spend at least 3 hours on the surface to recharge batteries. Once a U-boat detects an aircraft, it will immediately submerge to avoid detection. U-boats travel west-east from their ports using a shortest path to the Atlantic open waters, and back. U-boats leave port with 30 days of supplies, and return to port with no supplies left. Every U-boat that enters the North Atlantic has a 0.25 probability of being refueled and re-supplied and remaining on-station for an additional 30 days. We do not explicitly model tanker U-boats, capturing their effects with this 25% chance in increased on-station time.

V. Methodology

5.1. Game Theory Application

Historically, both the Allies and the Germans tested different operational strategies with the intent to gain an advantage over the other. The Allies had a search strategy which defined how they partitioned their search effort between daytime and nighttime hours. Likewise, the Germans followed a surfacing strategy which dictated the period of the day when they were allowed to surface. For example, a pure nighttime strategy allowed only nighttime surfacing for the U-boats, and was employed at one time by Germany (McCue, 1990:26). Eventually, the Allied operations researchers advocated a "balanced force" concept and their search effort "was distributed so that an hour spent on the surface at night was as dangerous for the U-boat as an hour spent on the surface in the daytime" (McCue, 1990:77-78).

On the contrary, U-boats wished to evade the aircraft. Logically, one would think the easiest means to accomplish this would be to surface either when the aircraft are not searching, or during times when it less dangerous to surface (such as night). However, this logic could be exploited by an adversary by committing all search efforts to the chosen surfacing period. Therefore, U-boats began to surface during both day and night hours, forcing the Allied search efforts to continue around the clock, and "making Allied concentration on any one part of the day fruitless" (McCue, 1990:78).

How then, did the Allies' and German's competing strategies affect one another? Were there strategies which were dominant and always produced better results than the alternatives? Was there the existence of a game theoretic equilibrium point, or saddle point, that the opponents drifted towards? What if strategies could not change, and conversely, what if they could? These questions and other characteristics are investigated with our agent-based model extended to incorporate game theory.

5.1.1. Formulation

Using concepts from game theory, the Bay of Biscay conflict can be depicted as a competitive game with two "players." These players are defined as the collective Allied search forces and the collective of German U-boats. Both sides are considered rational players because, as McCue states, "we may cite historical evidence as to Donitz's calculating rationality…[crucially], Donitz and the Allies operated according to the same objective function: merchant vessel sinkings" (1990:18). This conflict is also a game without perfect information, as neither side knows the exact strategies of the other.

The objectives for this particular game are to maximize number of U-boats detected by the Allied aircraft while the Germans desire to minimize detection. This makes the game a zero-sum game, for it will be construed that the Allies gained the number of detections, while the Germans "lost" the number of detections. The number of U-boat detections was chosen as the objective, rather than the number of U-boats killed (destroyed) for two reasons. First, the Allies sometimes were not able to immediately detect if they had killed a U-boat after dropping munitions. Waddington states that "it usually took at least some weeks… [to] arrive at a fairly firm estimate of the result of any attack, so that analyses of tactics could not be entirely up-to-date" (Waddington, 1973:169). Second, the number of U-boats killed was dependent on more than just the search and surfacing strategies of both sides; the probability of killing a U-boat after it was detected was very dependent on technology and other factors. In fact, during 1943, the probability of kill given detection was 0.40 by day and 0.11 by night.

The Allies had two pure search strategies available to them: search only by day, and search only by night. Likewise, the Germans had two pure surfacing strategies: surface only by day, and surface only by night. However, when one formulates the game to contain mixed strategies, we are able to partition the search efforts and surfacing policies into mixed strategies with varying probabilities.

Let α_1 be the Allied pure strategy of daytime search, and α_2 be the pure strategy of nighttime search. Then,

$$\begin{aligned}
&\mathrm{A_S} \equiv \text{Allied search strategy, such that } A_S \in (x_1\alpha_1, x_2\alpha_2) \\
&x_1, x_2 \in [0,1] \\
&x_1 + x_2 = 1
\end{aligned}$$

where x_1 is the percentage of search effort during daytime and x_2 is the percentage of search effort during nighttime.

Similarly, let β_1 be the Allied pure strategy of daytime search, and β_2 be the pure strategy of nighttime search. Then,

$$\begin{aligned}
&\mathrm{B_S} \equiv \text{U - Boat surfacing strategy, such that } B_S \in (y_1\beta_1, y_2\beta_2) \\
&y_1, y_2 \in [0,1] \\
&y_1 + y_2 = 1
\end{aligned}$$

where y_1 is the percentage of surfacing during daytime and y_2 is the percentage of surfacing during nighttime. Since the α's and β's are fixed, for simplification of nomenclature, the mixed strategies will be written from here on as (x_1, x_2) and (y_1, y_2). For example, an Allied search strategy of (0.7, 0.3) signifies 70% daytime searching and 30% nighttime searching.

This formulation closely resembles what is known as a *Colonel Blotto game*. A Colonel Blotto game, first defined by Borel (1938), is a two-person zero-sum game where both opposing players have *n* independent battlefields in which to distribute their forces, without knowing their opponent's distribution. This game involves the partitioning of resources to maximize some objective. This game could be considered a Colonel Blotto game if one thinks of daytime and nighttime as two independent "battlefields" and the choice to partition the Allied search aircraft and the U-boats surfacing as the strategies.

A typical game is analyzed by creating a game payoff matrix, similar to that in Figure 4, to represent each player's strategies and the resulting payoffs. The payoffs can either be values, or some deterministic function. These are mathematically examined to determine dominant strategies or equilibrium points. However, a combat campaign, like the Bay of Biscay campaign, is extremely complex with many probabilistic features. In order to determine the payoffs for this game, the simulation was run with replications at different design points, with the eventual goal of creating a response surface to be modeled and analyzed.

	B_1	B_2	B_3	…	B_N
A_1	P_{11}	P_{12}	P_{13}	…	P_{1N}
A_2	P_{21}	P_{22}	P_{23}	…	P_{2N}
A_3	P_{31}	P_{32}	P_{33}	…	P_{3N}
:	:	:	:	:	:
A_M	P_{M1}	P_{M2}	P_{M3}	…	P_{MN}

Figure 4. Typical Game Payoff Matrix

5.2. Scenarios and Run Design

The objective of this research is to explore the effects that different strategies have upon the number of U-boat detections. Three different scenarios are explored. For the first scenario, both the Allies and the Germans have “fixed” strategies, such that over the course of the simulation, these strategies do not change. The landscape of the results is explored using response surface methodology. For the second scenario, one side is allowed to adapt their strategy while the other side’s strategy remains fixed. These results are compared to those of the first scenario. Finally, scenario three looks at instances where both sides can adapt their strategies.

5.2.1. Model Modifications

The use of strategies, particularly any pure strategies, required that strict guidance for both the bombers and the U-boats be added to the baseline model’s code. For the Allied bombers, a schedule was implemented to distribute the takeoff times and search efforts between day and night hours. For the U-boats, conditional statements were created to ensure day and night surfacing events of every U-boat abided by the overall surfacing strategy. Additionally, stricter rules were applied to the model for the cases of pure strategies. For a detailed explanation of these model modifications, refer to Appendix A.

Another modification of the model was an algorithm enabling either player (or both simultaneously) to adapt their strategies during the simulation, based on perceived conditions. The algorithm was designed to be simple, use only data available to each side, and take little time to run. The algorithm exploited past strategies in the form of a strategy average, the current strategy, and a projected strategy based only on the current strategy setting and the immediate results. These three factors were combined in a weighted equation to produce the new strategy setting. For a more detailed explanation of the adaptation algorithm and equations, refer to Appendix A.

For scenarios one and two, the simulation modeled the time period April 1943 through September 1943. Historically, this time period contained no major technology changes, stabilizing the probability of kill given detection (daytime $P_{K|D} = 0.41$, nighttime $P_{K|D} = 0.11$).

5.2.2. Scenario One

This scenario required fixed strategies for both players. A 2^3 full-factorial design was used to explore the spectrum of strategies. The three settings chosen for each player's strategies are:

Aircraft: (1, 0), (0.5, 0.5), (0, 1)

U-boats: (1, 0), (0.5, 0.5), (0, 1)

Table 1 reflects the nine design points:

Table 1. Scenario One Design Points

Design Point	Allied Search Strategy	U-Boat Surfacing Strategy
1	(1, 0)	(1, 0)
2	(1, 0)	(0.5, 0.5)
3	(1, 0)	(0, 1)
4	(0, 1)	(0, 1)
5	(0.5, 0.5)	(1, 0)
6	(0.5, 0.5)	(0.5, 0.5)
7	(0.5, 0.5)	(0, 1)
8	(0, 1)	(1, 0)
9	(0, 1)	(0.5, 0.5)

Each design point was replicated 20 times, with 12 months of simulated warm-up time to distribute the U-boats, followed by 6 months of simulated time for data collection.

5.2.3. Scenario Two

This scenario required one player with a fixed strategy, and one player with an adaptive strategy. Adaptation of strategy occurred each month of the simulated timeframe. The nine design points in Table 1 were used, with the same number of replications and run lengths. This scenario was divided into two sets: Set A allowed only the Allied search aircraft to adapt, while Set B allowed only the U-boat's surfacing to adapt.

5.2.4. Scenario Three

This scenario gave both players adaptive strategies. In order to investigate this scenario, three design points were chosen at the following settings:

Table 2. Scenario Three Design Points

Design Point	Allied Search Strategy - Start	U-Boat Surfacing Strategy - Start
1	(1, 0)	(0, 1)
2	(1, 0)	(1, 0)
3	(0.5, 0.5)	(0.5, 0.5)

This case uses just three points since no matter what starting strategies are employed, over time the adaptation by both sides will yield similar strategies. These three points also represent interesting initial conditions: pure opposing strategies, pure matched strategies, and matched mixed strategies. A longer period of data collection was necessary to investigate if and how the adaptive strategies stabilize in the long run. For this case, adaptation of strategy occurred every month, with each run again involving 20 replications, 12 months of warm-up, but 12 months of data collection.

VI. Analysis and Results

6.1. Scenario One

6.1.1. Initial Hypothesis

A first look at this game scenario was accomplished by setting up a simplified game payoff matrix. We can logically state that at the two design points where opposing pure strategies occur, no U-boats will be detected. If Allies are only searching by day, and the U-boats are only surfacing at night, or vice versa, no U-boats should be found. Similarly, at the points where the two sides have matching pure strategies, large numbers of U-boats should be found. And at all other points, some U-boats should be detected (refer to Table 3).

Table 3. Scenario 1 Game Payoff Matrix

	B_s				
A_s	(0, 1)	…	(0.5, 0.5)	…	(1, 0)
(0,1)	Many		Some		0
⋮					
(0.5, 0.5)	Some		Some		Some
⋮					
(1, 0)	0		Some		Many

This initial analysis would appear to indicate that an equilibrium point could exist for this scenario, lying somewhere in the middle.

6.1.2. Actual Results

The actual empirical results are listed in Table 4 below, which shows the average number of U-boat detections (over the six month time period and 20 replications).

Table 4. Average Number of U-Boat Detections

Design Point	Allied Search Strategy	U-Boat Surfacing Strategy	Average Number U-Boat Detections
1	(1, 0)	(1, 0)	275.1
2	(1, 0)	(0.5, 0.5)	233.45
3	(1, 0)	(0, 1)	0
4	(0, 1)	(0, 1)	746.75
5	(0.5, 0.5)	(1, 0)	77.45
6	(0.5, 0.5)	(0.5, 0.5)	273.8
7	(0.5, 0.5)	(0, 1)	146.2
8	(0, 1)	(1, 0)	0
9	(0, 1)	(0.5, 0.5)	374.65

To perform the analysis using response surface methodology, all the data was analyzed using SAS JMP statistical software, Version 5.5. All data can be found in Appendix B. A response surface model was fit (using JMP), producing a model with an R^2 of 0.948 and an adjusted R^2 of 0.947 (Table 5):

Table 5. RSM Model Results

Summary of Fit

R Square	0.948501
R Square Adjusted	0.947021
Root Mean Square Error	50.56183
Mean of Response	236.3778
Observations	180

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	5	8192859.6	1638572	640.9439
Error	174	444830.7	2556	Prob > F
C. Total	179	8637690.3		<.0001

Lack of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	3	360068.31	120023	242.1344
Pure Error	171	84762.40	496	Prob > F
Total Error	174	444830.71		<.0001
				Max RSq
				0.9902

The model is shown in Equation 1:

$$z = 690.5077 + 423.27x_1^2 - 1138.475x_1 + 1021.85x_1y_1 - 345.53y_1^2 - 345.525y_1 \quad (1)$$

where x_1 is the aircraft day strategy, y_1 the U-boat day strategy, and z is the number of U-boat detections. Recall that x_2, the aircraft night strategy, is just $1 - x_1$. The same goes for y_2, the U-boat night strategy ($y_2 = 1 - y_1$).

Figure 5 profiles the contour plot, with the aircraft daytime strategy on the x-axis and the U-boat daytime strategy on the y-axis (the nighttime strategies are just the inverse). The contour plot depicts an existing equilibrium point for the mixed strategies.

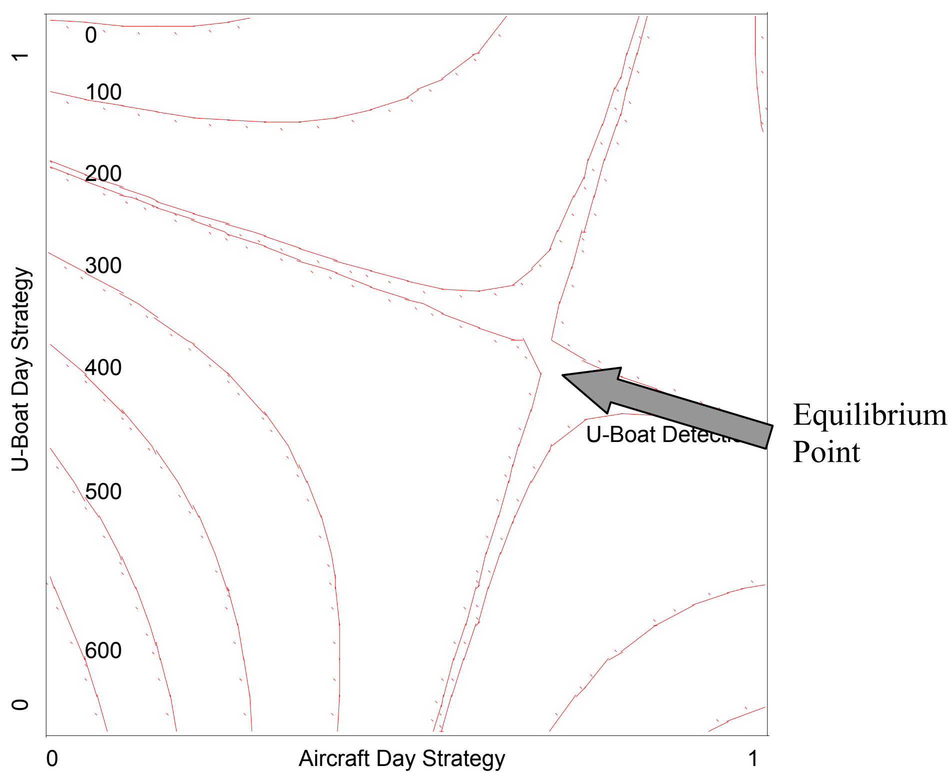


Figure 5. Contour Plot of RSM Model (Scenario 1)

A three-dimensional view of the response surface clearly shows the saddle point (Figure 6):

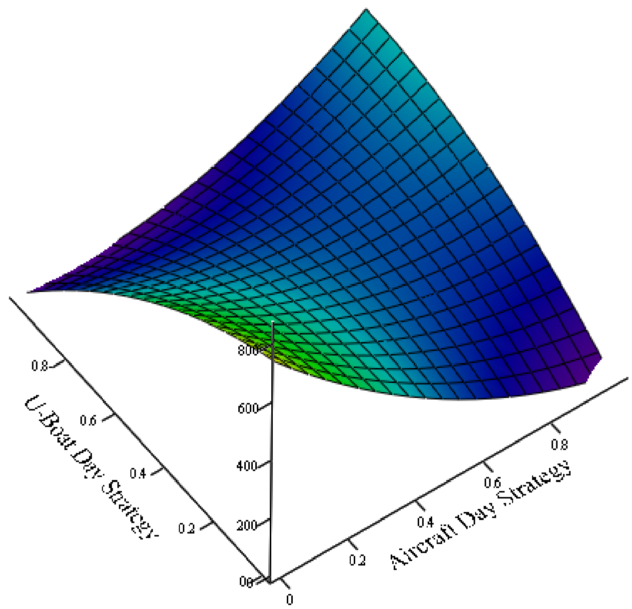


Figure 6. 3D View of Response Surface (Scenario 1)

The location of the equilibrium point is an aircraft strategy of (0.699, 0.301) and U-boat strategy of (0.534, 0.466). At this point, the Allies would be searching almost 70% during the day, and only 30% at night, while the U-boats surface almost evenly day and night. At first glance, this appears illogical since historically it was more difficult to detect a U-boat during the nighttime, thus the Allied equilibrium point should have a greater nighttime search percentage than the day. However, the results correspond to history.

During the months from April 1943 through September 1943, at the latitude and longitude of the Bay of Biscay, nighttime only accounts for approximately 8 hours, or 33%, of a 24-hour day. Essentially, at the equilibrium point the Allies are dedicating 70% of the search effort to match the approximate 66% time of daylight, and 30% search effort for the 33% time of night. This accurately matches what actually happened, since, as stated previously, the Allies adopted the balanced force concept to make it equally dangerous for U-boats to surface during both day and night. “Great emphasis was placed on the necessity for a properly balanced force, capable of attacking throughout the twenty-four hours” (Waddington, 1973:232). Likewise, our U-boat strategy at the equilibrium point mirrors history by allowing surfacing almost equally between day and night, to force the Allies to search around the clock.

6.2. Scenario Two

6.2.1. Initial Hypotheses

If one of the two players is allowed to adapt their strategy over time while the other strategy remains fixed, the logical conclusion is that the adapting player will change strategies to benefit themselves. For example, enabling the search aircraft to adapt strategies should enable them to determine U-boats’ surfacing strategy, and to nearly match it to find the most U-boats. Likewise, enabling the U-boats to adapt would allow them to surface when the least number of aircraft are searching, in order to minimize detections. The effect of this would be that the response surface found in Scenario One would level out and the equilibrium point should disappear.

6.2.2. Actual Results – Set A

For the case allowing the aircraft to adapt search strategies, Table 6 shows the average number of U-boat detections, and compares them to the results from Scenario One (Table 4):

Table 6. Results for Aircraft Strategy Adaptation

Design Point	Allied Search Strategy - Start	Average Search Strategy - End	U-Boat Surfacing Strategy	Average U-Boat Detections - Scenario 1	Average U-Boat Detections - Scenario 2A	Change
1	(1, 0)	(0.925, 0.075)	(1, 0)	275.1	103.7	Decrease
2	(1, 0)	(0.532, 0.468)	(0.5, 0.5)	233.45	101.7	Decrease
3	(1, 0)	(0.182, 0.818)	(0, 1)	0	171.7	Increase
4	(0, 1)	(0.076, 0.924)	(0, 1)	746.75	225	Decrease
5	(0.5, 0.5)	(0.889, 0.111)	(1, 0)	77.45	101.15	Increase
6	(0.5, 0.5)	(0.49, 0.51)	(0.5, 0.5)	273.8	105.4	Decrease
7	(0.5, 0.5)	(0.116, 0.884)	(0, 1)	146.2	210.45	Increase
8	(0, 1)	(0.816, 0.184)	(1, 0)	0	91.6	Increase
9	(0, 1)	(0.467, 0.533)	(0.5, 0.5)	374.65	112.7	Decrease

The average ending search strategies for the aircraft show that only after six updates (six months), the strategies are very close to the U-boat's surfacing strategies. The adaptation algorithm worked properly to hone in on the German's strategies in order to maximize the number of U-boat detections. To demonstrate, two graphs showing the adaptation of the aircraft's strategies over time are displayed in Figure 7 and Figure 8. Figure 7 shows the day strategies for all 20 replications for design point 9, while Figure 8 shows the mean day strategy for design point 9.

A response surface model was fit to the results. The model had an adjusted R^2 of 0.927. JMP did find an equilibrium point for the model at an aircraft strategy of (0.538, 0.462) and U-boat strategy of (0.787, 0.213), but the contour plot of the model (seen in Figure 9) shows that the surface has leveled out considerably. It is expected that given more updates (months), this equilibrium point would soon disappear. One must also realize that the model is built using the *starting* strategies and not the strategies after adaptation. The equation for the model is below (Equation 2):

$$z = 290.55 - 18.4x_1^2 + x_1 + 65.4x_1y_1 + 176y_1^2 - 279.57y_1 \qquad (2)$$

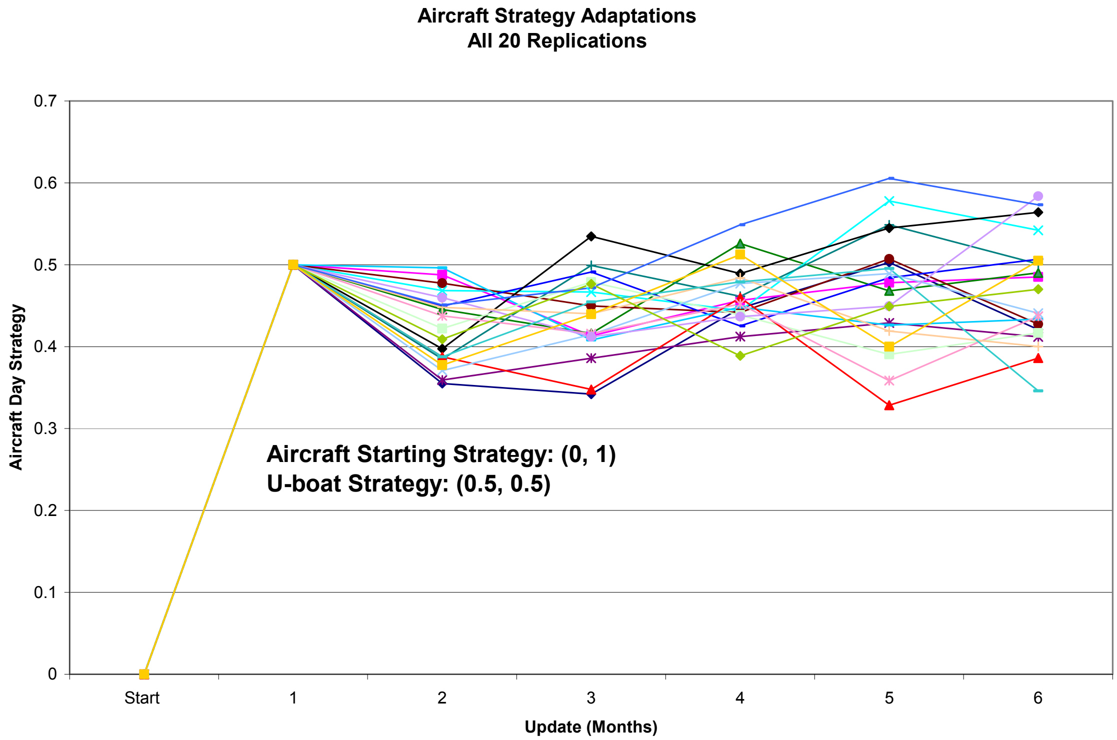


Figure 7. Aircraft Strategy Adaptations (20 replications) for Design Point 9

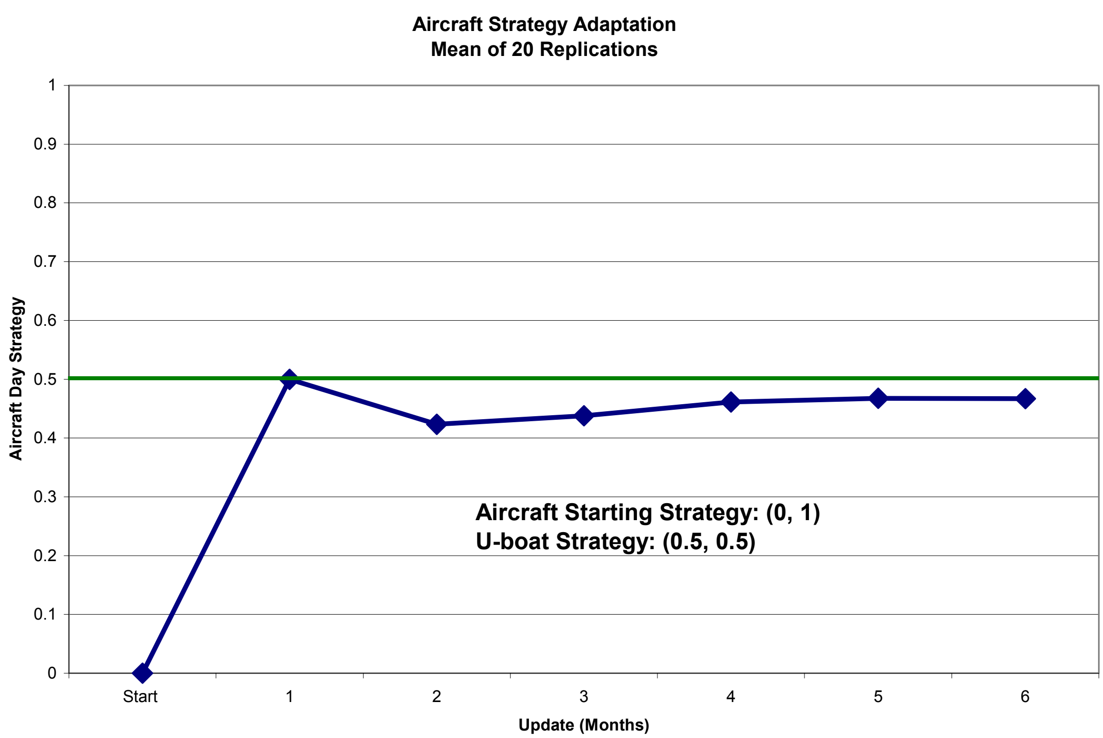


Figure 8. Mean Aircraft Strategy Adaptations for Design Point 9

In four of the design points there are increases in the average number of detections. Two of these cases are the design points with the opposing strategies. The other two cases occur for when the aircraft search strategy starts as (0.5, 0.5), and adapts toward a U-boat pure strategy.

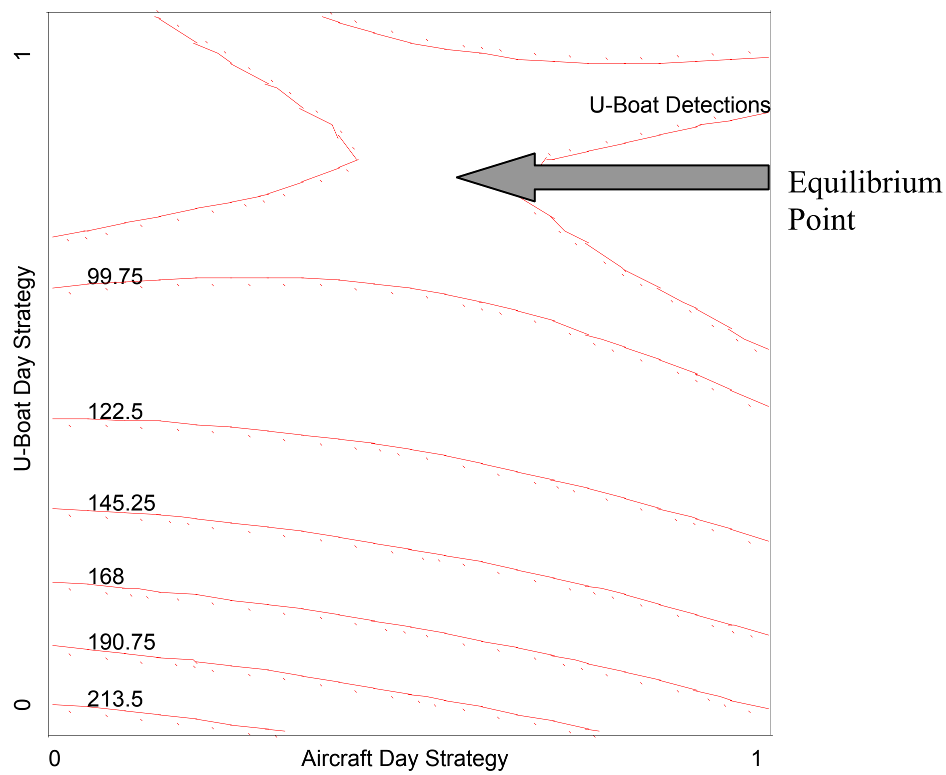


Figure 9. Contour Plot of RSM Model (Scenario 2A)

However, there is a decrease in the average number of detections in the remaining five cases. The reason lies within the logic behind the adaptation algorithm itself. First, the aircraft's algorithm, taking into account opponent adaptation, never permits the aircraft to reach or maintain a pure strategy. The reasoning behind this was that if the aircraft do not fly during one part of the day, then how would they know if the U-boats were surfacing during that period? The algorithm automatically kicks a pure strategy out to a mixed strategy of (0.5, 0.5), and adapts from there. For instance, at design point 1, the aircrafts strategy initially adapts away from the U-boats strategy, unlike Scenario One which continuously matches. This results in fewer U-boats being found over the 6-month period. Secondly, the algorithm has only had six updates in which to adapt. It would be expected that given more time, the results for the design points that had decreasing detection averages would eventually either result in increases or at least get close enough that there would be no statistical difference between the two averages.

6.2.3. Actual Results – Set B

For the case allowing the U-boats to adapt surfacing strategies, the following table shows the average number of U-boat detections, and compares them to the results from Scenario One:

Table 7. Results for U-Boat Strategy Adaptation

Design Point	Allied Search Strategy	U-Boat Surfacing Strategy - Start	Average Surfacing Strategy - End	Average U-Boat Detections - Scenario 1	Average U-Boat Detections - Scenario 2B	Change
1	(1, 0)	(1, 0)	(0.013, 0.987)	275.1	22.1	Decrease
2	(1, 0)	(0.5, 0.5)	(0, 1)	233.45	13.3	Decrease
3	(1, 0)	(0, 1)	(0, 1)	0	0	No Change
4	(0, 1)	(0, 1)	(1, 0)	746.75	27.95	Decrease
5	(0.5, 0.5)	(1, 0)	(0.568, 0.432)	77.45	97.45	Increase
6	(0.5, 0.5)	(0.5, 0.5)	(0.367, 0.633)	273.8	110.95	Decrease
7	(0.5, 0.5)	(0, 1)	(0.374, 0.626)	146.2	102.9	Decrease
8	(0, 1)	(1, 0)	(1, 0)	0	0	No Change
9	(0, 1)	(0.5, 0.5)	(0.950, 0.050)	374.65	99.5	Decrease

The results in Table 7 show that the U-boat's algorithm did very well at adapting away from the aircraft's strategies. Figures 10 and 11 show the day strategies for all 20 replications for design point 9, and the mean day strategy across all replications. Table 7 also reveals that this algorithm performed better under these circumstances than did the aircraft's. All but one of the original design points experienced a reduction or no change in the average number of detections. (The no changes were design points 3 and 8, opposing pure strategies that had also originally found zero U-boats; a decrease from zero is not possible).

Again, the results were used to fit a response surface model. The model had an adjusted R^2 of 0.6399, signifying more variation within the data than the previous two models. However, JMP did not find an equilibrium point; in fact there is maximum located at aircraft strategy (0.448, 0.552) and U-boat strategy (0.476, 0.524). The contour plot of the model (Figure 12) also shows a maximum on the surface. When aircraft have pure strategies, the U-boats quickly adapt towards the opposite pure strategy. When the aircraft have a (0.5, 0.5) mixed strategy, they cannot find a safe pure strategy to move towards and end up fluctuating around the (0.5, 0.5) strategy. Significantly more U-boats are found at these design points (points 5, 6, and 7), forcing the non-linearity and local maximum. The equation for the model is shown here (Equation 3):

$$z = 33.415 - 306.5x_1^{\,2} + 275.82x_1 + 50.05x_1y_1 - 131.4y_1^{\,2} + 131.4y_1 \quad (3)$$

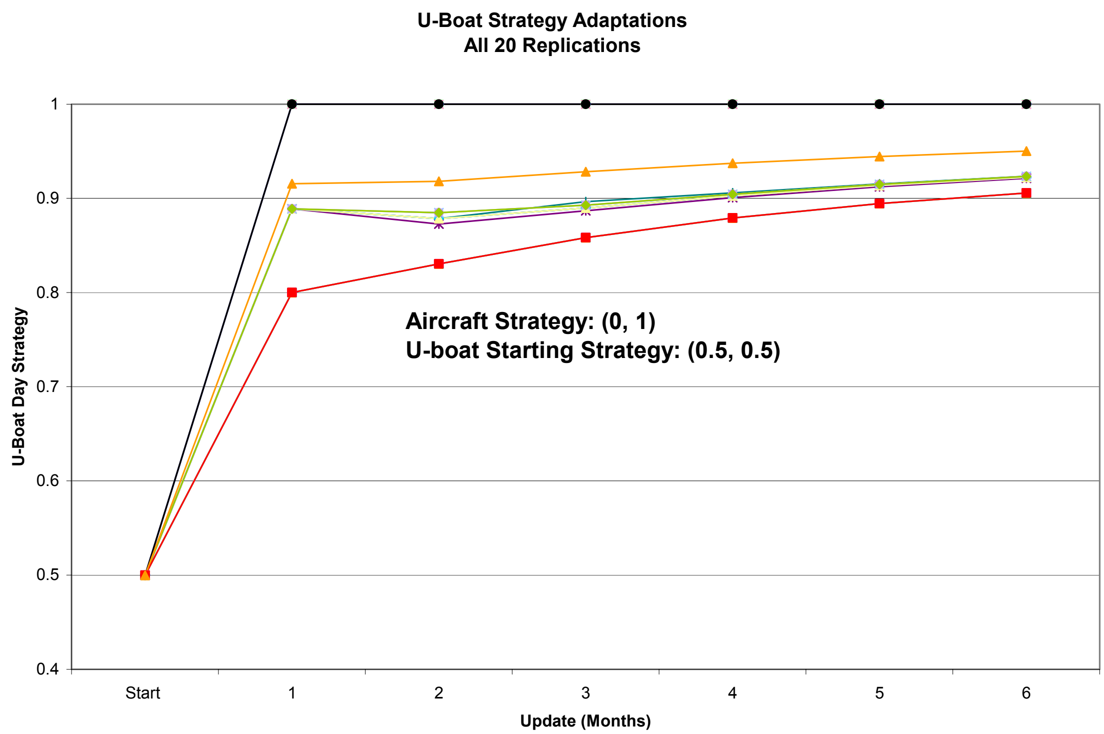


Figure 10. U-Boat Strategy Adaptations (20 replications)

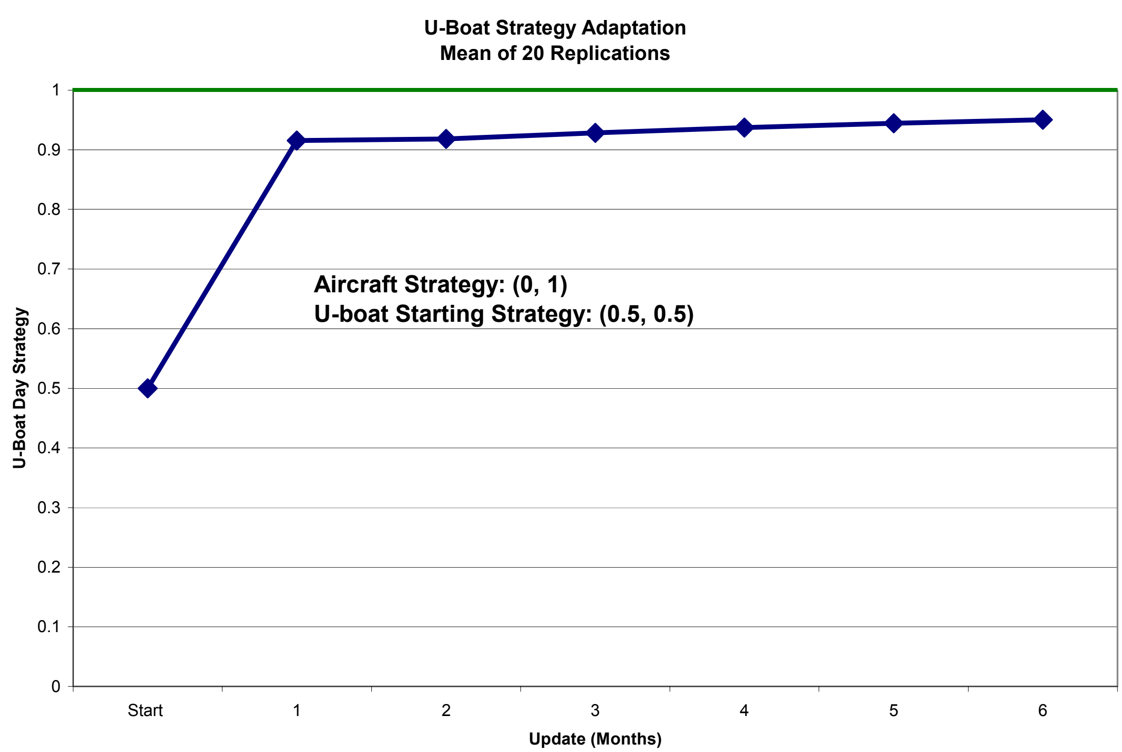


Figure 11. U-Boat Strategy Adaptations (Mean)

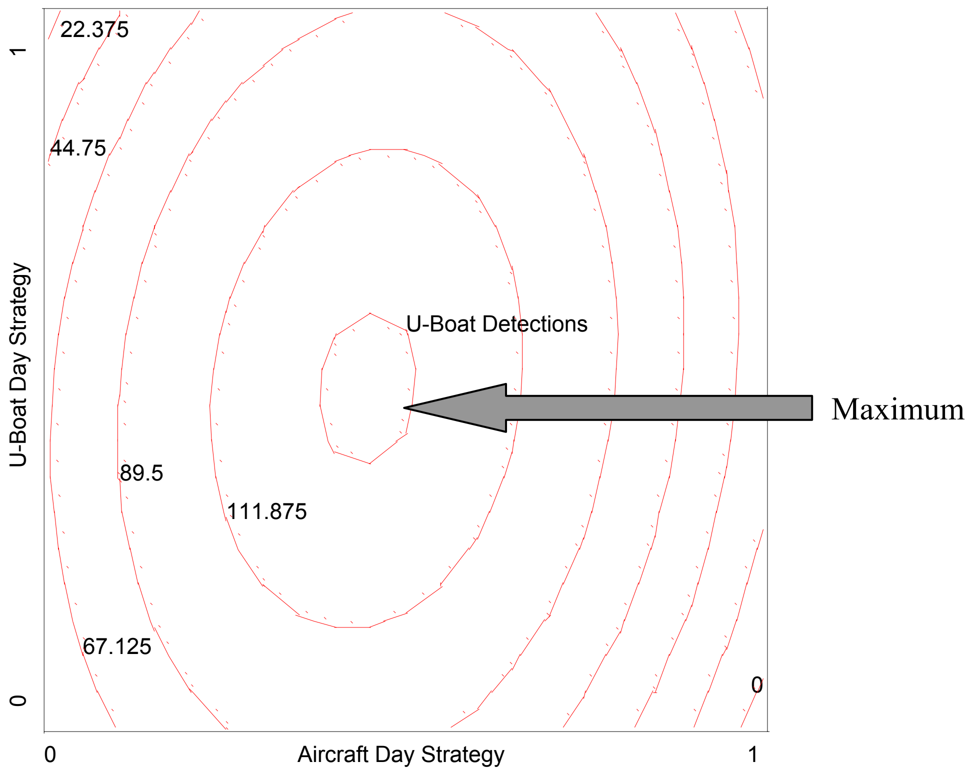


Figure 12. Contour Plot of RSM Model (Scenario 2B)

6.3. Scenario Three

6.3.1. Initial Hypothesis

If both players have the capability to adapt their strategies, in the long run these strategies should stabilize around some equilibrium. In doing so, the average number of U-boat detections should be similar.

6.3.2. Actual Results

The starting and ending average strategies, along with the average number of U-boat detections are provided in Table 8:

Table 8. Results for Two-Player Adaptation

Design Point	Allied Search Strategy - Start	Allied Search Strategy - End	U-Boat Surfacing Strategy - Start	U-Boat Surfacing Strategy - End	Average Number U-Boat Detections
1	(1, 0)	(0.542, 0.458)	(0, 1)	(0.164, 0.836)	183.75
2	(1, 0)	(0.625, 0.375)	(1, 0)	(0.327, 0.673)	180.45
3	(0.5, 0.5)	(0.522, 0.478)	(0.5, 0.5)	(0.259, 0.741)	182.6

The numbers show that each side ended up with a mixed strategy. The average ending strategies for the aircraft are very close to each other, while for the U-boats these ending strategies were all weighted towards nighttime surfacing. Looking at the graphs for each design point (Figures 13, 14, and 15, respectively), it is fairly obvious that the strategies have stabilized near an equilibrium strategy. Although from design point to design point there seems to be differences in what this equilibrium strategy might be, this may be an artifact of the initial starting conditions for each strategy or due to the stochastic nature of the system.

Furthermore, the average numbers of U-boat detections over these 12-month periods are very close. So close, in fact, that a both a student's t-test and a Tukey-Kramer test for differences in means at an alpha level of 0.05 result in no significant statistical differences among the averages. See Figure 16 for the SAS JMP analysis.

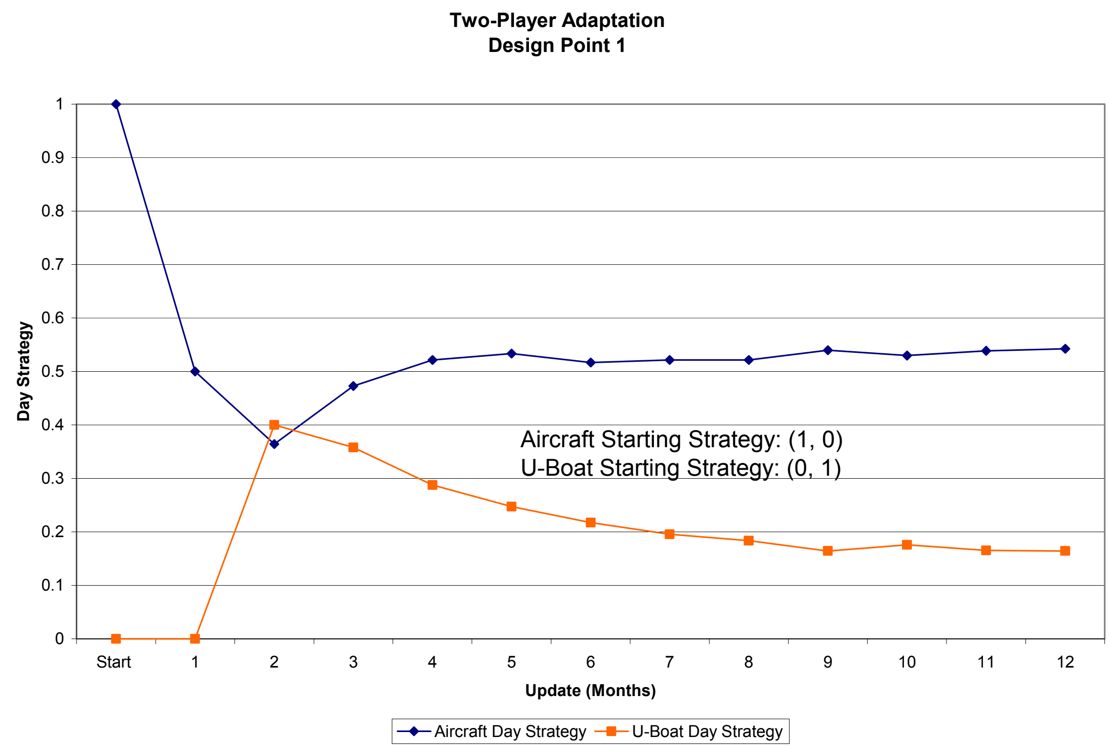


Figure 13. Two-Player Adaptation, Design Point 1

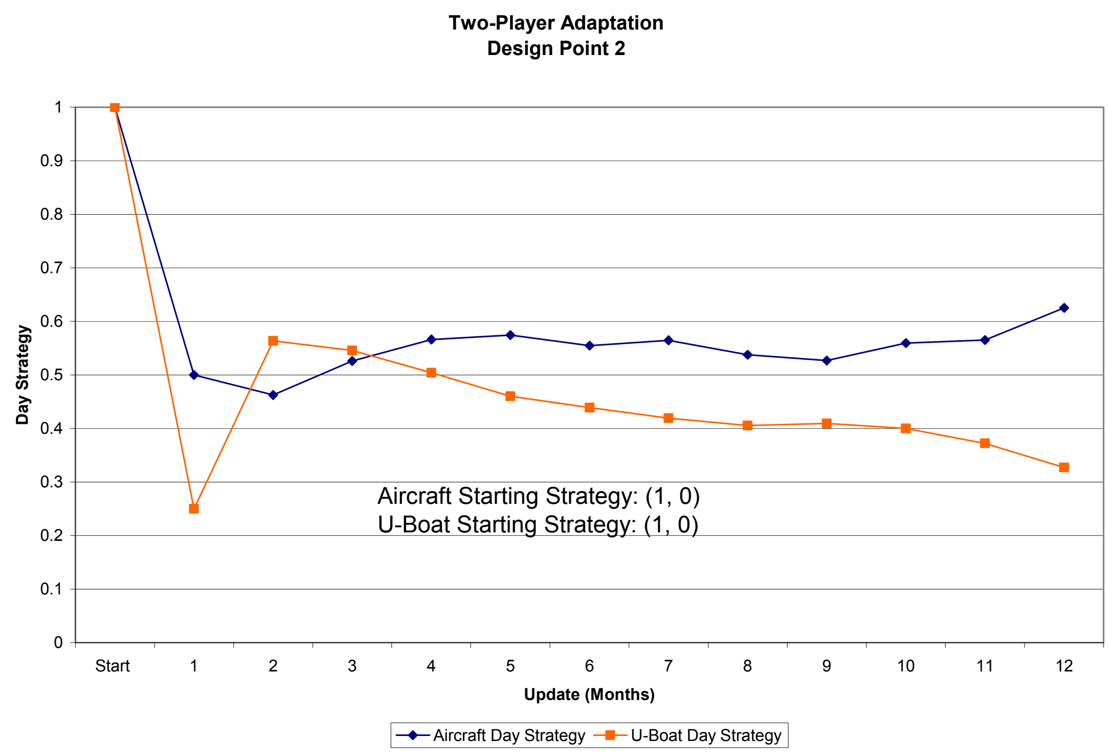


Figure 14. Two-Player Adaptation, Design Point 2

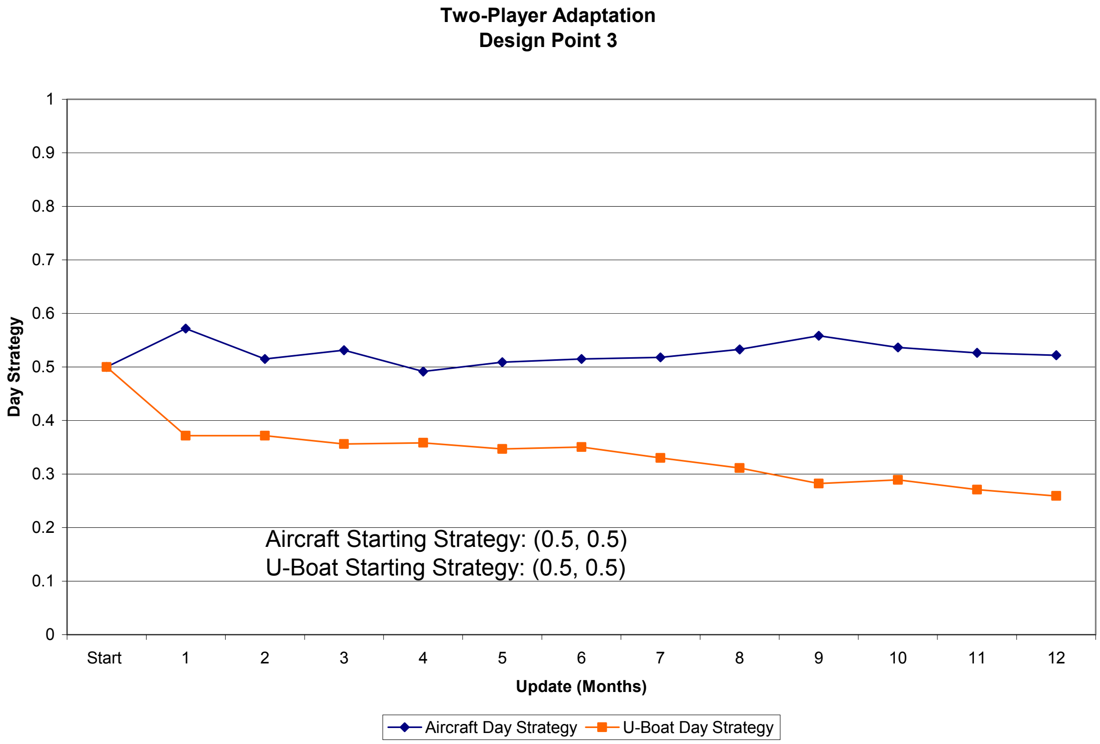


Figure 15. Two-Player Adaptation, Design Point 3

<u>Alpha = 0.05</u>

Comparisons for each pair using Student's t:

t	Alpha
2.00247	0.05

Levels not connected by same letter are significantly different:

Level		Mean
1	A	183.75000
3	A	182.60000
2	A	180.45000

Comparisons for all pairs using Tukey-Kramer HSD:

q*	Alpha
2.40642	0.05

Levels not connected by same letter are significantly different:

Level		Mean
1	A	183.75000
3	A	182.60000
2	A	180.45000

Figure 16. Means Comparison Test Output (SAS JMP)

VII. Conclusion

This research used an agent-based simulation to investigate game-theoretic strategies of a combat scenario. The research showed that for mixed strategies an equilibrium point did exist, and that the strategies at this point concurred with historical actions. The capability of one side to adapt strategies during the campaign changes the shape of the strategies' response surface, eventually eliminating the equilibrium point altogether. This adapting player is also able to take advantage of the other player's fixed strategies, and increase their payoff in certain areas. Finally, two adapting players will, given time, end up at some level of equilibrium, possibly dependent on their initial conditions. Fluctuations in these levels could be due to the stochastic nature of the complex system, or a lack in adaptation efficiency. Overall, this research has demonstrated that a properly modeled agent-based system is a viable means to analyze the game-theoretic properties of a complex system.

7.1. Recommendations for Future Work

Future research should focus on extending this work in three areas. First, extend this work to accommodate increased data collection times for one-sided and two-sided adaptation. This will garner insight into the disappearance of equilibrium points for the one-sided case, and the determination of where strategy equilibrium exists for the two-sided case. Also, look at running the fixed strategy case again for the remaining six months out of the year, to see if the changing daylight hours still impact the equilibrium point.

Second, improve the adaptation algorithm to increase its adaptation effectiveness. Only a minimal number of test runs were accomplished to assess its performance characteristics. There may be a way to optimize the weighted functions to improve performance. More powerful methods to determine new strategies would allow for quicker adaptation.

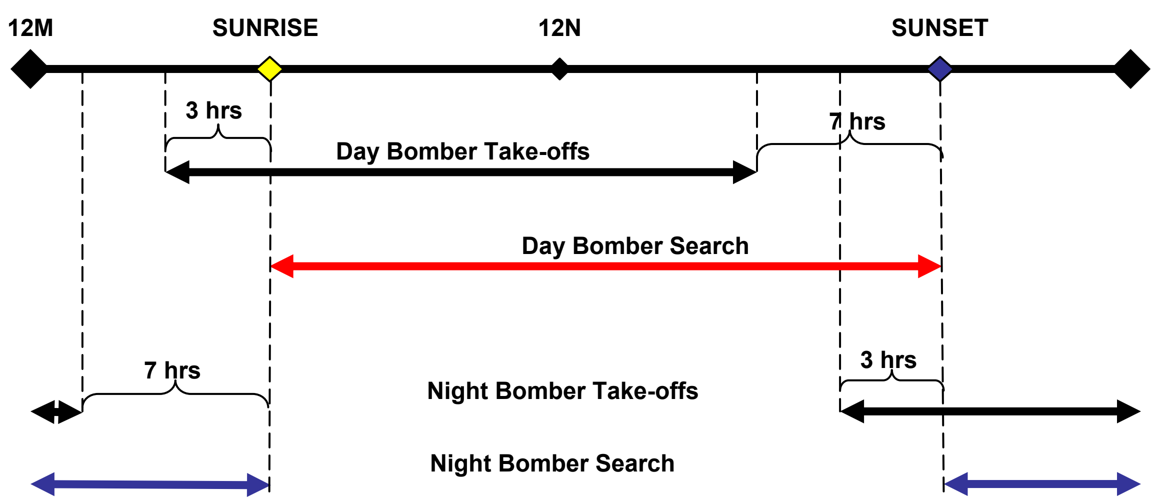
Finally, investigate other factors that were held constant during the simulation, such as technology levels, probability of kills, and alternative tactics such as "wolf-packs," and determine their effects on the strategies. Enable adaptation of technology/counter-technology over time to alter probability of detection and probability of kill. Look at alternative tactics for both sides, like U-boat "wolf-packs," the policy of crossing the Bay in groups for safety; surfaced crossings where U-boats stay on the surface as much as possible to travel at maximum speed and minimize time in the Bay; or an Allied group search, where aircraft search in close proximity to each other and can radio in other aircraft to attack detected U-boats. Finally, enable some form of communication between agents in order to promote learning. The Allies could use this to facilitate a type of priority search, where the place emphasis on searching those grids with larger numbers of detected U-boats. The Germans could use communication to travel by avoidance routing, by which they avoid those areas of the Bay which have had the most U-boat detections or kills.

Appendix A: Code Modifications for Game Theory

Bomber Code Modifications

To enable search aircraft to be partitioned between day and night searches for a given Allied search strategy, a method was devised to assign bombers to search at either day or night. In the JAVA code, a bomber was designated either a "day bomber" or a "night bomber." A "day bomber" could only search for U-boats during daylight hours, while a "night bomber" could only search during nighttime hours. Additionally, a rule was imposed that prevented aircraft to search after their assigned time periods. For example, day bombers could not search beyond sunset; once this "transition point" was reached, day bombers would immediately stop searching and return to base. The same rule held for night bombers and the sunrise transition point.

To ensure day bombers and night bombers took advantage of their entire time window for searching, a schedule was created to coordinate daily takeoff times. This schedule allowed bombers to takeoff up to three hours before their starting transition point and travel to their assigned search grid. Only when the transition point to their assigned search time occurred were the bombers allowed to start searching. The last possible bomber takeoff time was scheduled for seven hours before the next transition point. This allowed a bomber taking off at that time three hours to travel to its search grid and four hours of search time before the next transition point occurred, forcing the bomber to stop searching and return to base. Between the earliest and latest possible takeoff times for the bombers, the actual takeoff time was scheduled using a uniform random number draw between these two bounds. The takeoff schedule concept is shown below:



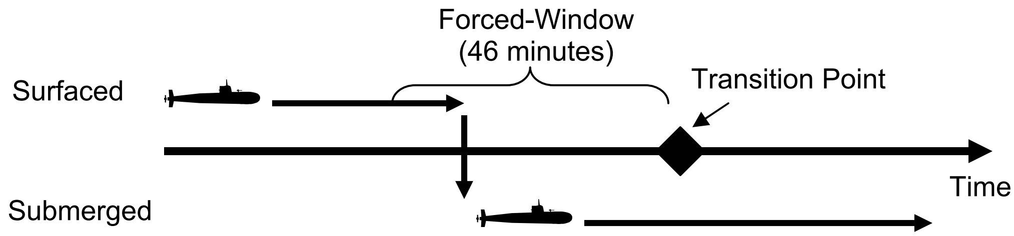
For a pure strategy of daytime searching, all aircraft were designated as day bombers. Likewise, a pure strategy of nighttime searching meant all aircraft were designated as night bombers. When a mixed strategy was assigned, the aircraft in the squadron were designated either as day bombers or night bombers according to the partitioning of the mixed strategy. For example, if the day search strategy was 0.70, then 70% of the bombers would search during the daytime hours.

U-boat Code Modifications

To partition the U-boats between day and night surfacing, special code was added to the model. A uniform random number draw between 0 and 1 occurred each time a U-boat needed to surface. If this value corresponded to the probability of surfacing during that specific time period, and it occurred during said time period, the U-boat surfaced. If not, the U-boat waited for a set period of time before attempting to surface again. For example, if the probability of surfacing at night was 0.70, and the random number draw produced a value of 0.5432, and it was also the nighttime period, then the U-boat surfaced. Additionally, if the random number draw produced a value of 0.877, and it was the daytime period, then that U-boat also surfaced. A simple example of all possible results appears in the following table:

Case	**Random Number Draw**	**Night Surfacing Strategy**	**Time Period**	**Result**
1	0.5	0.7	Nighttime	Surface
2	0.9	0.7	Nighttime	Stay submerged
3	0.5	0.7	Daytime	Stay submerged
4	0.9	0.7	Daytime	Surface

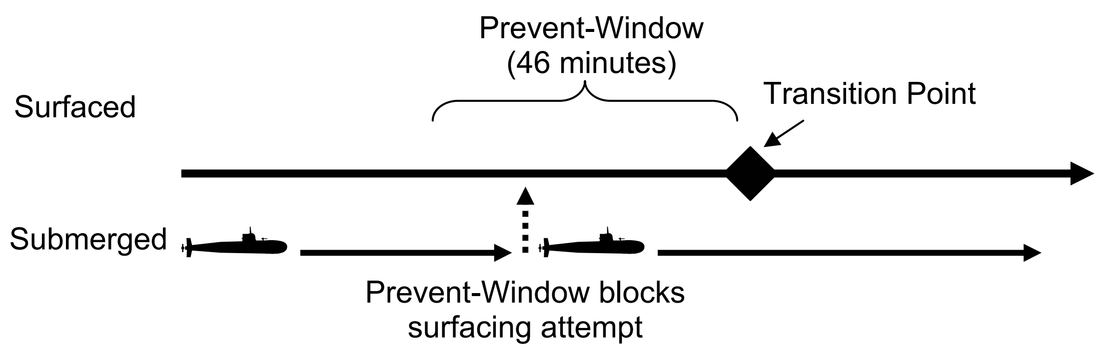
In the cases of pure strategies, strict measures were applied to ensure proper surfacing and timely submergence. These conditional statements looked for two particular situations and ensure the U-boats followed the logic for a pure strategy. First, if a U-boat had appropriately surfaced, but the time needed to recharge the batteries would force it to remain surfaced during the transition point to the next, and prohibited, time period, the conditional statement forced the U-boat to submerge when it entered a window of time before the transition point. This window of time is called the forced-window, and began 46 simulated minutes before a transition point, ensuring all updating U-boats on the surface would be identified and forced to submerge. This window was required due to the fact that in the model, U-boats update their state (position, submergence, etc.) in set increments of time. A window this large was chosen to ensure that every U-boat would update at least once within the time span of the window, and would submerge. Additionally, if a U-boat did not fully recharge its batteries, it could then travel a distance proportional to the actual time it spent on the surface. The following diagram details this forced-window condition.



Example Scenario:

- Pure nighttime surfacing only, current time period is nighttime
- U-boat is surfaced, next update is at 0416 hours
- Transition point (sunrise) is at 0430 hours
- Result: at next update (0416 hrs), U-boat will be in forced-window (0344 – 0430 hours); U-boat will submerge and continue travel

The second situation occurs if a U-boat is submerged during its allowable surfacing period, but is scheduled to surface right before the transition point. Another 46 minute window is established before the transition point to prevent the U-boat from surfacing. The purpose is to prevent U-boats from surfacing immediately before a transition point and remaining on the surface during the prohibited time period. This window is called the prevent-window. The following diagram details this condition.



Example Scenario:

- Pure nighttime surfacing only, current time period is nighttime
- U-boat is submerged, next update is at 0416 hours and will result in surfacing
- Transition point (sunrise) is at 0430 hours
- Result: at next update (0416 hrs), U-boat will be in prevent-window (0344-0430 hours); U-boat will remain submerged

It must be emphasized that both of these conditions and windows are only enforced when the U-boats are operating under pure surfacing strategies.

Adaptation Algorithm

The purpose of the adaptation algorithm within the model is to enable both the Allies and the U-boats to change their search or surfacing strategies over set time increments in order to optimize their outcomes. The adaptation mechanism was designed to meet certain criteria for performance. First, the algorithm was limited to only use statistics from the model as inputs. These statistics needed to reflect a measure of historical accuracy, such that they were a type of information available to decision makers during World War II. Second, the algorithm had to be easy to implement, but detailed to a degree to produce some means of efficient adaptation in strategy. Finally, the algorithm had to be flexible enough to enable reaction and adaptation to possible changes in an opponent's strategy, but simultaneously be stable enough so that the strategies were not wildly fluctuating over time.

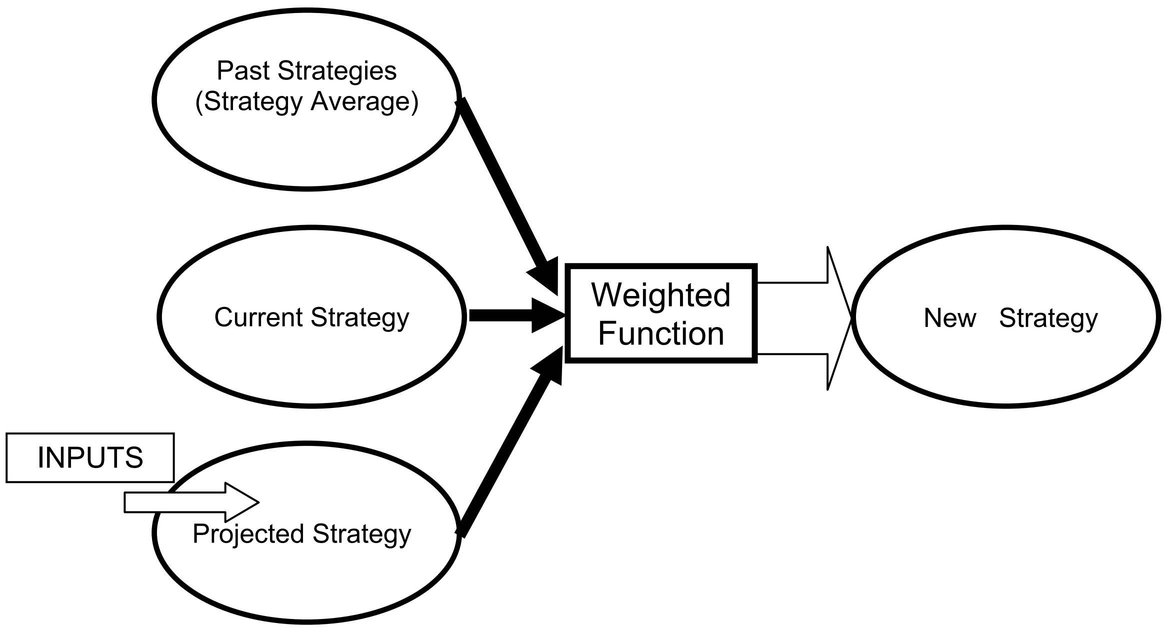
Within the JAVA model, a new Strategy class was created to house the adaptation algorithm. This class was responsible for collecting appropriate statistics for inputs, and after a new strategy had been determined, would update the strategies for all aircraft and/or U-boat agents. This class was structured such that only one side or the other, or both simultaneously, could change their strategy. The class also tracked these strategy changes over time and reported these levels with the normal model output.

The first step in the creation of the algorithm determined the type of information to be used for each side as inputs. For the Allies, it was simply the number of U-boat detections over the given time period. This is logical as a sighted U-boat was direct evidence that U-boats were surfacing during a particular time period. As stated in the main body of this thesis, often times the Allies did not have reliable or timely data regarding the destruction of a U-boat. So historically and logically, the only reliable measure of when the U-boats were surfacing is when aircraft detected surfaced U-boats.

For the Germans, the number of inputs for determining the aircraft search strategy tripled. The first input was the number of U-boat detections. Historically, U-boats could sometimes determine when aircraft had detected them, either based on the fact that the aircraft were currently attacking, or that their radar search receivers alerted them to incoming aircraft. The second input, closely related, was the number of aircraft sighted. If a U-boat detected a nearby aircraft, it would know the time period when that aircraft was searching. Finally, the third input was the number of U-boats killed. Admiral Donitz's staff kept in daily radio contact with all U-boats, so they immediately knew if one had been destroyed. Thus, it was known that aircraft were searching during the same time period that U-boats were being killed.

Inputs into Adaptation Algorithm	
Aircraft	**U-boats**
Number of U-boat detections	Number of U-boat detections
	Number of aircraft sightings
	Number of U-boats killed

Once the inputs for each side's adaptation algorithm were defined, the next step was to formulate a process to find a new strategy from this limited information. Besides these inputs, each side had available to them their previous strategies and the current operating strategy. These three pieces of information could be formulated and combined to result in a new strategy. The method by which this was accomplished is outlined in the diagram below.



A side's previous strategies were tracked and averaged together to form a representation of past strategies. This component of the new strategy would provide a necessary stability and prevent uncontrollable oscillations in strategy based on possible inputs that are misleading and do not reflect the opponent's true strategy. The current strategy should theoretically encapsulate the successful adaptation of strategy up to the present. And the projected strategy, determined solely by the inputs from the most recent time increment, gives an instantaneous snapshot of what the strategy should be to take advantage of the opposition's strategy. These three components are combined using a weighted function, resulting in a new strategy. The weights for this function used in the model were arbitrarily determined by the author to reflect logical considerations. These weights are shown here:

Allied Component	**Bomber Day Strategy Average**	**Current Bomber Strategy**	**Projected Bomber Strategy**
Weight	0.25	0.35	0.40
U-boat Component	**U-boat Day Strategy Average**	**Current U-boat Strategy**	**Projected U-boat Strategy**
Weight	0.25	0.35	0.40

The past strategies and the current strategy components were easily acquired. The complicated part was to transform the inputs into a projected strategy in some logical manner. The best way to illustrate this is to step through this process for one of the sides; the aircraft adaptation algorithm was selected for this.

First, a "day factor" is calculated using the following equation:

$$\text{Day Factor} = \frac{\text{Day U} - \text{boat Detections}/\text{Current Bomber Day Strategy}}{\text{Day U} - \text{boat Detections} + \text{Night U} - \text{boat Detections}}$$

A "night factor" is calculated in a similar fashion:

$$\text{Night Factor} = \frac{\text{Night U} - \text{boat Detections}/\text{Current Bomber Night Strategy}}{\text{Day U} - \text{boat Detections} + \text{Night U} - \text{boat Detections}}$$

These factors essentially represent the new projected day and night strategies, based only on the number of U-boat detections during a given time period. However, they must be scaled to each other, using this equation:

$$\text{Projected Day Strategy} = \frac{\text{Day Factor}}{\text{Day Factor} + \text{Night Factor}}$$

This projected day strategy is combined with two other components from above, the day strategy average and the current day strategy, within a function weighted as previously mentioned. This produces the new bomber day strategy. The new bomber night strategy is simply 1 – Projected Day Strategy, since the strategies must sum to one. The projected day strategy for the U-boats is calculated in a like manner, except there are three day and three night factors, one for each of the three inputs to the U-boat algorithm. The three day factors are combined into an overall day factor using a weighted sum, with the weights chosen by the author based upon the relevance of the backing input to the overall determination of strategy. For example, since the possibility of a U-boat sighting an aircraft was small, the aircraft detected day factor only receives a weight of 0.2:

$$\text{Day Factor} = (0.2 * \text{aircraft sighted day factor}) + (0.4 * \text{U - boats killed day factor}) + (0.4 * \text{U - boats detected day factor})$$

The overall U-boat night factor is calculated likewise.

Both sides have special considerations for their respective algorithm. For example, if the aircraft are operating at a pure search strategy, then how do the Allies know if any U-boats are surfacing at the other time period? This state tends to remain at a pure strategy, because no U-boats are being detected at the opposite time period, ever. In order to avoid this pitfall, code was added so that if the Allies are searching at a pure strategy, during the next strategy update the algorithm adjusts the strategy such that the aircraft are searching (0.5, 0.5) for the next time increment automatically. The aircraft algorithm, if no U-boats are detected for an entire time increment, automatically switches the day strategy and the night strategy to find a possible time that the U-boats may be exploiting for surfacing.

The main difference between the aircraft and the U-boat algorithms lies in purpose. The aircraft algorithm attempts to find and adapt towards the opponent's strategy, while the U-boat algorithm attempts to stay away from the opponent's strategy. As such, the U-boat algorithm has several unique features to encourage this behavior. For example, if there are no U-boat detections or U-boat kills for a time increment, then the U-boat strategy will not change at all. Also, based upon average numbers of U-boat detections and kills from Scenario One, a detection threshold and a kill threshold are in place so that if the current U-boat strategy is at or near a pure strategy, and the U-boat detections and kills are under these thresholds, then the U-boat strategy will not change for the next time increment.

Lessons Learned – JAVA, Multi-Agent Systems, and Windows

The biggest problem the author encountered during the coding and execution of the modified model was a problem resulting from the way JAVA executes on a Windows system. It was found that the run time for twenty replications at a design point would take anywhere from two to four days to accomplish. During this long run time, it was discovered that the central processing unit (CPU) of the computer would be operating between 94 and 100 percent. Over time, such as into the tenth replication, this extreme usage of the processor would eventually slow the computer down and the program would advance simulated time at a very slow pace.

After one of these runs at a design point with opposing pure strategies, results appeared which did not match up with what should have happened. For example, with opposing pure strategies, there should be no U-boat detections at all. However, the output reported differently, finding U-boats in many of the replications. After many weeks of painstaking scrutiny of the code looking for a phantom "bug" which may have caused this problem, the real problem was discovered.

In JAVA, every agent has its own thread, and the execution of a program with many threads is accomplished by *multithreading*. With a Windows Operating System, each thread is given a portion of execution time with the processor called a *quantum*, and this process is known as *timeslicing*. It was discovered that when the simulation was monitored during a run and paused every once in a while, the system never bogged down and no problems with the output data were seen. Thus it was hypothesized that when the CPU becomes overloaded with threads during the long runs, something happens to foul up the proper timeslicing and so some threads are given multiple turns before others get to have their turn. This explained the erroneous data.

To correct for this, code was added to the model that forced the model to pause for a set time (1 minute) between each replication. This pause allows the CPU to rest and for expended threads to be collected by the system and dumped. After implementation of the pause code, not only was there no more incorrect output, but the time in which a design point was accomplished decreased dramatically, from 2-4 days to 12-13 hours. It is a good tip for others modeling large complex systems to be aware of how your program is running on the computer and what the program is doing to the memory and the processor capabilities.

Appendix B: Data

Scenario 1:

Design Point	Strategy		Detections		
	Aircraft	U-boat	Total	Day	Night
1	(1, 0)	(1, 0)	273	273	0
1	(1, 0)	(1, 0)	239	239	0
1	(1, 0)	(1, 0)	280	280	0
1	(1, 0)	(1, 0)	252	252	0
1	(1, 0)	(1, 0)	277	277	0
1	(1, 0)	(1, 0)	289	289	0
1	(1, 0)	(1, 0)	305	305	0
1	(1, 0)	(1, 0)	275	275	0
1	(1, 0)	(1, 0)	262	262	0
1	(1, 0)	(1, 0)	272	272	0
1	(1, 0)	(1, 0)	284	284	0
1	(1, 0)	(1, 0)	275	275	0
1	(1, 0)	(1, 0)	283	283	0
1	(1, 0)	(1, 0)	259	259	0
1	(1, 0)	(1, 0)	292	292	0
1	(1, 0)	(1, 0)	297	297	0
1	(1, 0)	(1, 0)	252	252	0
1	(1, 0)	(1, 0)	263	263	0
1	(1, 0)	(1, 0)	294	294	0
1	(1, 0)	(1, 0)	279	279	0
2	(1, 0)	(0.5, 0.5)	212	212	0
2	(1, 0)	(0.5, 0.5)	241	241	0
2	(1, 0)	(0.5, 0.5)	231	231	0
2	(1, 0)	(0.5, 0.5)	223	223	0
2	(1, 0)	(0.5, 0.5)	194	194	0
2	(1, 0)	(0.5, 0.5)	211	211	0
2	(1, 0)	(0.5, 0.5)	233	233	0
2	(1, 0)	(0.5, 0.5)	227	227	0
2	(1, 0)	(0.5, 0.5)	224	224	0
2	(1, 0)	(0.5, 0.5)	228	228	0
2	(1, 0)	(0.5, 0.5)	245	245	0
2	(1, 0)	(0.5, 0.5)	245	245	0
2	(1, 0)	(0.5, 0.5)	246	246	0
2	(1, 0)	(0.5, 0.5)	234	234	0
2	(1, 0)	(0.5, 0.5)	228	228	0
2	(1, 0)	(0.5, 0.5)	236	236	0
2	(1, 0)	(0.5, 0.5)	257	257	0
2	(1, 0)	(0.5, 0.5)	248	248	0
2	(1, 0)	(0.5, 0.5)	265	265	0
2	(1, 0)	(0.5, 0.5)	241	241	0

Design Point	Strategy		Detections		
	Aircraft	U-boat	Total	Day	Night
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
3	(1, 0)	(0, 1)	0	0	0
4	(0, 1)	(0, 1)	781	0	781
4	(0, 1)	(0, 1)	709	0	709
4	(0, 1)	(0, 1)	754	0	754
4	(0, 1)	(0, 1)	807	0	807
4	(0, 1)	(0, 1)	792	0	792
4	(0, 1)	(0, 1)	688	0	688
4	(0, 1)	(0, 1)	733	0	733
4	(0, 1)	(0, 1)	741	0	741
4	(0, 1)	(0, 1)	723	0	723
4	(0, 1)	(0, 1)	717	0	717
4	(0, 1)	(0, 1)	795	0	795
4	(0, 1)	(0, 1)	806	0	806
4	(0, 1)	(0, 1)	779	0	779
4	(0, 1)	(0, 1)	726	0	726
4	(0, 1)	(0, 1)	818	0	818
4	(0, 1)	(0, 1)	772	0	772
4	(0, 1)	(0, 1)	659	0	659
4	(0, 1)	(0, 1)	786	0	786
4	(0, 1)	(0, 1)	651	0	651
4	(0, 1)	(0, 1)	698	0	698

Design Point	Strategy		Detections		
	Aircraft	U-boat	Total	Day	Night
5	(0.5, 0.5)	(1, 0)	78	78	0
5	(0.5, 0.5)	(1, 0)	79	79	0
5	(0.5, 0.5)	(1, 0)	75	75	0
5	(0.5, 0.5)	(1, 0)	76	76	0
5	(0.5, 0.5)	(1, 0)	88	88	0
5	(0.5, 0.5)	(1, 0)	78	78	0
5	(0.5, 0.5)	(1, 0)	81	81	0
5	(0.5, 0.5)	(1, 0)	68	68	0
5	(0.5, 0.5)	(1, 0)	84	84	0
5	(0.5, 0.5)	(1, 0)	72	72	0
5	(0.5, 0.5)	(1, 0)	75	75	0
5	(0.5, 0.5)	(1, 0)	84	84	0
5	(0.5, 0.5)	(1, 0)	72	72	0
5	(0.5, 0.5)	(1, 0)	73	73	0
5	(0.5, 0.5)	(1, 0)	70	70	0
5	(0.5, 0.5)	(1, 0)	88	88	0
5	(0.5, 0.5)	(1, 0)	86	86	0
5	(0.5, 0.5)	(1, 0)	64	64	0
5	(0.5, 0.5)	(1, 0)	76	76	0
5	(0.5, 0.5)	(1, 0)	82	82	0
6	(0.5, 0.5)	(0.5, 0.5)	263	142	121
6	(0.5, 0.5)	(0.5, 0.5)	273	150	123
6	(0.5, 0.5)	(0.5, 0.5)	270	146	124
6	(0.5, 0.5)	(0.5, 0.5)	230	127	103
6	(0.5, 0.5)	(0.5, 0.5)	274	137	137
6	(0.5, 0.5)	(0.5, 0.5)	275	160	115
6	(0.5, 0.5)	(0.5, 0.5)	263	147	116
6	(0.5, 0.5)	(0.5, 0.5)	300	166	134
6	(0.5, 0.5)	(0.5, 0.5)	299	162	137
6	(0.5, 0.5)	(0.5, 0.5)	275	166	109
6	(0.5, 0.5)	(0.5, 0.5)	273	149	124
6	(0.5, 0.5)	(0.5, 0.5)	260	152	108
6	(0.5, 0.5)	(0.5, 0.5)	242	129	113
6	(0.5, 0.5)	(0.5, 0.5)	284	150	134
6	(0.5, 0.5)	(0.5, 0.5)	262	159	103
6	(0.5, 0.5)	(0.5, 0.5)	301	186	115
6	(0.5, 0.5)	(0.5, 0.5)	270	150	120
6	(0.5, 0.5)	(0.5, 0.5)	284	144	140
6	(0.5, 0.5)	(0.5, 0.5)	292	164	128
6	(0.5, 0.5)	(0.5, 0.5)	286	160	126

Design Point	Strategy		Detections		
	Aircraft	U-boat	Total	Day	Night
7	(0.5, 0.5)	(0, 1)	121	0	121
7	(0.5, 0.5)	(0, 1)	163	0	163
7	(0.5, 0.5)	(0, 1)	159	0	159
7	(0.5, 0.5)	(0, 1)	150	0	150
7	(0.5, 0.5)	(0, 1)	151	0	151
7	(0.5, 0.5)	(0, 1)	164	0	164
7	(0.5, 0.5)	(0, 1)	140	0	140
7	(0.5, 0.5)	(0, 1)	129	0	129
7	(0.5, 0.5)	(0, 1)	131	0	131
7	(0.5, 0.5)	(0, 1)	134	0	134
7	(0.5, 0.5)	(0, 1)	152	0	152
7	(0.5, 0.5)	(0, 1)	154	0	154
7	(0.5, 0.5)	(0, 1)	155	0	155
7	(0.5, 0.5)	(0, 1)	151	0	151
7	(0.5, 0.5)	(0, 1)	151	0	151
7	(0.5, 0.5)	(0, 1)	141	0	141
7	(0.5, 0.5)	(0, 1)	144	0	144
7	(0.5, 0.5)	(0, 1)	163	0	163
7	(0.5, 0.5)	(0, 1)	125	0	125
7	(0.5, 0.5)	(0, 1)	146	0	146
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0
8	(0, 1)	(1, 0)	0	0	0

Design Point	Strategy		Detections		
	Aircraft	U-boat	Total	Day	Night
9	(0, 1)	(0.5, 0.5)	383	0	383
9	(0, 1)	(0.5, 0.5)	365	0	365
9	(0, 1)	(0.5, 0.5)	358	0	358
9	(0, 1)	(0.5, 0.5)	361	0	361
9	(0, 1)	(0.5, 0.5)	378	0	378
9	(0, 1)	(0.5, 0.5)	379	0	379
9	(0, 1)	(0.5, 0.5)	342	0	342
9	(0, 1)	(0.5, 0.5)	385	0	385
9	(0, 1)	(0.5, 0.5)	372	0	372
9	(0, 1)	(0.5, 0.5)	447	0	447
9	(0, 1)	(0.5, 0.5)	411	0	411
9	(0, 1)	(0.5, 0.5)	332	0	332
9	(0, 1)	(0.5, 0.5)	424	0	424
9	(0, 1)	(0.5, 0.5)	362	0	362
9	(0, 1)	(0.5, 0.5)	372	0	372
9	(0, 1)	(0.5, 0.5)	396	0	396
9	(0, 1)	(0.5, 0.5)	375	0	375
9	(0, 1)	(0.5, 0.5)	337	0	337
9	(0, 1)	(0.5, 0.5)	384	0	384
9	(0, 1)	(0.5, 0.5)	330	0	330

Scenario 2A:

Design Point	Strategy Aircraft	Strategy U-boat	Detections Total
1	(1, 0)	(1, 0)	114
1	(1, 0)	(1, 0)	102
1	(1, 0)	(1, 0)	100
1	(1, 0)	(1, 0)	89
1	(1, 0)	(1, 0)	107
1	(1, 0)	(1, 0)	118
1	(1, 0)	(1, 0)	111
1	(1, 0)	(1, 0)	104
1	(1, 0)	(1, 0)	94
1	(1, 0)	(1, 0)	101
1	(1, 0)	(1, 0)	90
1	(1, 0)	(1, 0)	110
1	(1, 0)	(1, 0)	97
1	(1, 0)	(1, 0)	105
1	(1, 0)	(1, 0)	109
1	(1, 0)	(1, 0)	113
1	(1, 0)	(1, 0)	97
1	(1, 0)	(1, 0)	103
1	(1, 0)	(1, 0)	108
1	(1, 0)	(1, 0)	102
2	(1, 0)	(0.5, 0.5)	120
2	(1, 0)	(0.5, 0.5)	90
2	(1, 0)	(0.5, 0.5)	126
2	(1, 0)	(0.5, 0.5)	93
2	(1, 0)	(0.5, 0.5)	110
2	(1, 0)	(0.5, 0.5)	111
2	(1, 0)	(0.5, 0.5)	105
2	(1, 0)	(0.5, 0.5)	94
2	(1, 0)	(0.5, 0.5)	97
2	(1, 0)	(0.5, 0.5)	109
2	(1, 0)	(0.5, 0.5)	90
2	(1, 0)	(0.5, 0.5)	97
2	(1, 0)	(0.5, 0.5)	84
2	(1, 0)	(0.5, 0.5)	119
2	(1, 0)	(0.5, 0.5)	90
2	(1, 0)	(0.5, 0.5)	93
2	(1, 0)	(0.5, 0.5)	100
2	(1, 0)	(0.5, 0.5)	103
2	(1, 0)	(0.5, 0.5)	113
2	(1, 0)	(0.5, 0.5)	90

Design Point	Strategy Aircraft	Strategy U-boat	Detections Total
3	(1, 0)	(0, 1)	147
3	(1, 0)	(0, 1)	161
3	(1, 0)	(0, 1)	163
3	(1, 0)	(0, 1)	177
3	(1, 0)	(0, 1)	146
3	(1, 0)	(0, 1)	172
3	(1, 0)	(0, 1)	187
3	(1, 0)	(0, 1)	166
3	(1, 0)	(0, 1)	173
3	(1, 0)	(0, 1)	185
3	(1, 0)	(0, 1)	183
3	(1, 0)	(0, 1)	178
3	(1, 0)	(0, 1)	175
3	(1, 0)	(0, 1)	172
3	(1, 0)	(0, 1)	152
3	(1, 0)	(0, 1)	165
3	(1, 0)	(0, 1)	205
3	(1, 0)	(0, 1)	182
3	(1, 0)	(0, 1)	160
3	(1, 0)	(0, 1)	185
4	(0, 1)	(0, 1)	212
4	(0, 1)	(0, 1)	220
4	(0, 1)	(0, 1)	246
4	(0, 1)	(0, 1)	212
4	(0, 1)	(0, 1)	245
4	(0, 1)	(0, 1)	216
4	(0, 1)	(0, 1)	225
4	(0, 1)	(0, 1)	210
4	(0, 1)	(0, 1)	252
4	(0, 1)	(0, 1)	182
4	(0, 1)	(0, 1)	208
4	(0, 1)	(0, 1)	249
4	(0, 1)	(0, 1)	212
4	(0, 1)	(0, 1)	206
4	(0, 1)	(0, 1)	225
4	(0, 1)	(0, 1)	245
4	(0, 1)	(0, 1)	241
4	(0, 1)	(0, 1)	240
4	(0, 1)	(0, 1)	218
4	(0, 1)	(0, 1)	236

Design	Strategy		Detections
Point	Aircraft	U-boat	Total
5	(0.5, 0.5)	(1, 0)	98
5	(0.5, 0.5)	(1, 0)	107
5	(0.5, 0.5)	(1, 0)	98
5	(0.5, 0.5)	(1, 0)	104
5	(0.5, 0.5)	(1, 0)	104
5	(0.5, 0.5)	(1, 0)	95
5	(0.5, 0.5)	(1, 0)	104
5	(0.5, 0.5)	(1, 0)	95
5	(0.5, 0.5)	(1, 0)	117
5	(0.5, 0.5)	(1, 0)	92
5	(0.5, 0.5)	(1, 0)	93
5	(0.5, 0.5)	(1, 0)	102
5	(0.5, 0.5)	(1, 0)	97
5	(0.5, 0.5)	(1, 0)	106
5	(0.5, 0.5)	(1, 0)	108
5	(0.5, 0.5)	(1, 0)	109
5	(0.5, 0.5)	(1, 0)	113
5	(0.5, 0.5)	(1, 0)	93
5	(0.5, 0.5)	(1, 0)	95
5	(0.5, 0.5)	(1, 0)	93
6	(0.5, 0.5)	(0.5, 0.5)	102
6	(0.5, 0.5)	(0.5, 0.5)	104
6	(0.5, 0.5)	(0.5, 0.5)	106
6	(0.5, 0.5)	(0.5, 0.5)	114
6	(0.5, 0.5)	(0.5, 0.5)	110
6	(0.5, 0.5)	(0.5, 0.5)	98
6	(0.5, 0.5)	(0.5, 0.5)	91
6	(0.5, 0.5)	(0.5, 0.5)	85
6	(0.5, 0.5)	(0.5, 0.5)	106
6	(0.5, 0.5)	(0.5, 0.5)	126
6	(0.5, 0.5)	(0.5, 0.5)	104
6	(0.5, 0.5)	(0.5, 0.5)	105
6	(0.5, 0.5)	(0.5, 0.5)	117
6	(0.5, 0.5)	(0.5, 0.5)	109
6	(0.5, 0.5)	(0.5, 0.5)	97
6	(0.5, 0.5)	(0.5, 0.5)	101
6	(0.5, 0.5)	(0.5, 0.5)	102
6	(0.5, 0.5)	(0.5, 0.5)	79
6	(0.5, 0.5)	(0.5, 0.5)	101
6	(0.5, 0.5)	(0.5, 0.5)	151

Design	Strategy		Detections
Point	Aircraft	U-boat	Total
7	(0.5, 0.5)	(0, 1)	210
7	(0.5, 0.5)	(0, 1)	218
7	(0.5, 0.5)	(0, 1)	184
7	(0.5, 0.5)	(0, 1)	259
7	(0.5, 0.5)	(0, 1)	202
7	(0.5, 0.5)	(0, 1)	233
7	(0.5, 0.5)	(0, 1)	189
7	(0.5, 0.5)	(0, 1)	198
7	(0.5, 0.5)	(0, 1)	222
7	(0.5, 0.5)	(0, 1)	241
7	(0.5, 0.5)	(0, 1)	212
7	(0.5, 0.5)	(0, 1)	189
7	(0.5, 0.5)	(0, 1)	195
7	(0.5, 0.5)	(0, 1)	215
7	(0.5, 0.5)	(0, 1)	223
7	(0.5, 0.5)	(0, 1)	231
7	(0.5, 0.5)	(0, 1)	209
7	(0.5, 0.5)	(0, 1)	211
7	(0.5, 0.5)	(0, 1)	200
7	(0.5, 0.5)	(0, 1)	168
8	(0, 1)	(1, 0)	94
8	(0, 1)	(1, 0)	88
8	(0, 1)	(1, 0)	103
8	(0, 1)	(1, 0)	81
8	(0, 1)	(1, 0)	92
8	(0, 1)	(1, 0)	95
8	(0, 1)	(1, 0)	99
8	(0, 1)	(1, 0)	94
8	(0, 1)	(1, 0)	89
8	(0, 1)	(1, 0)	83
8	(0, 1)	(1, 0)	77
8	(0, 1)	(1, 0)	101
8	(0, 1)	(1, 0)	87
8	(0, 1)	(1, 0)	99
8	(0, 1)	(1, 0)	89
8	(0, 1)	(1, 0)	92
8	(0, 1)	(1, 0)	78
8	(0, 1)	(1, 0)	101
8	(0, 1)	(1, 0)	104
8	(0, 1)	(1, 0)	86

Design	Strategy		Detections
Point	Aircraft	U-boat	Total
9	(0, 1)	(0.5, 0.5)	116
9	(0, 1)	(0.5, 0.5)	101
9	(0, 1)	(0.5, 0.5)	105
9	(0, 1)	(0.5, 0.5)	109
9	(0, 1)	(0.5, 0.5)	125
9	(0, 1)	(0.5, 0.5)	116
9	(0, 1)	(0.5, 0.5)	109
9	(0, 1)	(0.5, 0.5)	116
9	(0, 1)	(0.5, 0.5)	121
9	(0, 1)	(0.5, 0.5)	115
9	(0, 1)	(0.5, 0.5)	105
9	(0, 1)	(0.5, 0.5)	114
9	(0, 1)	(0.5, 0.5)	110
9	(0, 1)	(0.5, 0.5)	118
9	(0, 1)	(0.5, 0.5)	102
9	(0, 1)	(0.5, 0.5)	117
9	(0, 1)	(0.5, 0.5)	101
9	(0, 1)	(0.5, 0.5)	115
9	(0, 1)	(0.5, 0.5)	123
9	(0, 1)	(0.5, 0.5)	116

Scenario 2B:

Design Point	Strategy		Detections
	Aircraft	U-boat	Total
1	(1, 0)	(1, 0)	25
1	(1, 0)	(1, 0)	14
1	(1, 0)	(1, 0)	16
1	(1, 0)	(1, 0)	18
1	(1, 0)	(1, 0)	25
1	(1, 0)	(1, 0)	20
1	(1, 0)	(1, 0)	21
1	(1, 0)	(1, 0)	19
1	(1, 0)	(1, 0)	19
1	(1, 0)	(1, 0)	16
1	(1, 0)	(1, 0)	12
1	(1, 0)	(1, 0)	19
1	(1, 0)	(1, 0)	17
1	(1, 0)	(1, 0)	97
1	(1, 0)	(1, 0)	18
1	(1, 0)	(1, 0)	12
1	(1, 0)	(1, 0)	14
1	(1, 0)	(1, 0)	23
1	(1, 0)	(1, 0)	14
1	(1, 0)	(1, 0)	23
2	(1, 0)	(0.5, 0.5)	17
2	(1, 0)	(0.5, 0.5)	11
2	(1, 0)	(0.5, 0.5)	20
2	(1, 0)	(0.5, 0.5)	17
2	(1, 0)	(0.5, 0.5)	12
2	(1, 0)	(0.5, 0.5)	16
2	(1, 0)	(0.5, 0.5)	10
2	(1, 0)	(0.5, 0.5)	13
2	(1, 0)	(0.5, 0.5)	8
2	(1, 0)	(0.5, 0.5)	19
2	(1, 0)	(0.5, 0.5)	15
2	(1, 0)	(0.5, 0.5)	16
2	(1, 0)	(0.5, 0.5)	14
2	(1, 0)	(0.5, 0.5)	10
2	(1, 0)	(0.5, 0.5)	15
2	(1, 0)	(0.5, 0.5)	9
2	(1, 0)	(0.5, 0.5)	8
2	(1, 0)	(0.5, 0.5)	11
2	(1, 0)	(0.5, 0.5)	15
2	(1, 0)	(0.5, 0.5)	10

Design Point	Strategy		Detections
	Aircraft	U-boat	Total
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
3	(1, 0)	(0, 1)	0
4	(0, 1)	(0, 1)	23
4	(0, 1)	(0, 1)	34
4	(0, 1)	(0, 1)	22
4	(0, 1)	(0, 1)	33
4	(0, 1)	(0, 1)	33
4	(0, 1)	(0, 1)	24
4	(0, 1)	(0, 1)	22
4	(0, 1)	(0, 1)	25
4	(0, 1)	(0, 1)	27
4	(0, 1)	(0, 1)	23
4	(0, 1)	(0, 1)	36
4	(0, 1)	(0, 1)	28
4	(0, 1)	(0, 1)	30
4	(0, 1)	(0, 1)	34
4	(0, 1)	(0, 1)	26
4	(0, 1)	(0, 1)	28
4	(0, 1)	(0, 1)	37
4	(0, 1)	(0, 1)	25
4	(0, 1)	(0, 1)	24
4	(0, 1)	(0, 1)	25

Design Point	Strategy		Detections
	Aircraft	U-boat	Total
5	(0.5, 0.5)	(1, 0)	75
5	(0.5, 0.5)	(1, 0)	120
5	(0.5, 0.5)	(1, 0)	90
5	(0.5, 0.5)	(1, 0)	107
5	(0.5, 0.5)	(1, 0)	109
5	(0.5, 0.5)	(1, 0)	98
5	(0.5, 0.5)	(1, 0)	99
5	(0.5, 0.5)	(1, 0)	85
5	(0.5, 0.5)	(1, 0)	92
5	(0.5, 0.5)	(1, 0)	95
5	(0.5, 0.5)	(1, 0)	111
5	(0.5, 0.5)	(1, 0)	104
5	(0.5, 0.5)	(1, 0)	99
5	(0.5, 0.5)	(1, 0)	109
5	(0.5, 0.5)	(1, 0)	87
5	(0.5, 0.5)	(1, 0)	95
5	(0.5, 0.5)	(1, 0)	78
5	(0.5, 0.5)	(1, 0)	93
5	(0.5, 0.5)	(1, 0)	85
5	(0.5, 0.5)	(1, 0)	118
6	(0.5, 0.5)	(0.5, 0.5)	112
6	(0.5, 0.5)	(0.5, 0.5)	112
6	(0.5, 0.5)	(0.5, 0.5)	106
6	(0.5, 0.5)	(0.5, 0.5)	101
6	(0.5, 0.5)	(0.5, 0.5)	137
6	(0.5, 0.5)	(0.5, 0.5)	100
6	(0.5, 0.5)	(0.5, 0.5)	92
6	(0.5, 0.5)	(0.5, 0.5)	91
6	(0.5, 0.5)	(0.5, 0.5)	117
6	(0.5, 0.5)	(0.5, 0.5)	113
6	(0.5, 0.5)	(0.5, 0.5)	95
6	(0.5, 0.5)	(0.5, 0.5)	117
6	(0.5, 0.5)	(0.5, 0.5)	111
6	(0.5, 0.5)	(0.5, 0.5)	111
6	(0.5, 0.5)	(0.5, 0.5)	101
6	(0.5, 0.5)	(0.5, 0.5)	116
6	(0.5, 0.5)	(0.5, 0.5)	114
6	(0.5, 0.5)	(0.5, 0.5)	128
6	(0.5, 0.5)	(0.5, 0.5)	126
6	(0.5, 0.5)	(0.5, 0.5)	119

Design Point	Strategy		Detections
	Aircraft	U-boat	Total
7	(0.5, 0.5)	(0, 1)	66
7	(0.5, 0.5)	(0, 1)	110
7	(0.5, 0.5)	(0, 1)	90
7	(0.5, 0.5)	(0, 1)	103
7	(0.5, 0.5)	(0, 1)	111
7	(0.5, 0.5)	(0, 1)	114
7	(0.5, 0.5)	(0, 1)	107
7	(0.5, 0.5)	(0, 1)	106
7	(0.5, 0.5)	(0, 1)	91
7	(0.5, 0.5)	(0, 1)	111
7	(0.5, 0.5)	(0, 1)	115
7	(0.5, 0.5)	(0, 1)	112
7	(0.5, 0.5)	(0, 1)	102
7	(0.5, 0.5)	(0, 1)	120
7	(0.5, 0.5)	(0, 1)	81
7	(0.5, 0.5)	(0, 1)	111
7	(0.5, 0.5)	(0, 1)	114
7	(0.5, 0.5)	(0, 1)	89
7	(0.5, 0.5)	(0, 1)	104
7	(0.5, 0.5)	(0, 1)	101
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0
8	(0, 1)	(1, 0)	0

Design Point	Strategy		Detections
	Aircraft	U-boat	Total
9	(0, 1)	(0.5, 0.5)	159
9	(0, 1)	(0.5, 0.5)	156
9	(0, 1)	(0.5, 0.5)	151
9	(0, 1)	(0.5, 0.5)	179
9	(0, 1)	(0.5, 0.5)	159
9	(0, 1)	(0.5, 0.5)	9
9	(0, 1)	(0.5, 0.5)	142
9	(0, 1)	(0.5, 0.5)	10
9	(0, 1)	(0.5, 0.5)	150
9	(0, 1)	(0.5, 0.5)	21
9	(0, 1)	(0.5, 0.5)	13
9	(0, 1)	(0.5, 0.5)	143
9	(0, 1)	(0.5, 0.5)	158
9	(0, 1)	(0.5, 0.5)	18
9	(0, 1)	(0.5, 0.5)	167
9	(0, 1)	(0.5, 0.5)	14
9	(0, 1)	(0.5, 0.5)	8
9	(0, 1)	(0.5, 0.5)	15
9	(0, 1)	(0.5, 0.5)	153
9	(0, 1)	(0.5, 0.5)	165

Bibliography

Alpern, Steve and Shmuel Gal. "Searching for an Agent Who May or May Not Want to be Found," *Operations Research*, Vol. 50, No. 2: 311-323 (March-April 2002).

Baston, V. J. and F. A. Bostock. "A One-Dimensional Helicopter-Submarine Game," *Naval Research Logistics*, Vol. 36: 479-490 (1989).

Batty, Michael and Bin Jiang. *Multi-Agent Simulation: New Approaches to Exploring Space-Time Dynamics within GIS*. Working Paper 10. Center for Advanced Spatial Analysis, 1999.

Benkoski, Stanley J., Michael G. Monticino, and James R. Weisinger. "A Survey of the Search Theory Literature," *Naval Research Logistics*, Vol. 38: 469-494 (1991).

Bjorkman, Eileen A. *Could We Have Predicted the Outcome of Desert Storm? Improving Campaign Analysis Through Agent-Based Modeling*. Executive Research Report, Industrial College of the Armed Forces. Washington DC: National Defense University, 2000.

Borel, E., *Traite du calcul des probabilities et des ses applications, Applications des jeux de hazard*, Vol. IV, Fascicule 2, Gauthier-Villars, Paris France, 1938.

Brandstein, Alfred G. and Gary E. Horne. "Data Farming: A Meta-Technique for Research in the 21st Century," in *Maneuver Warfare Science 1998*. Eds. F. G. Hoffman and Gary E. Horne. Quantico VA: Marine Corps Combat Development Command, 1998.

Champagne, Lance E. and Raymond R. Hill. "Analysis of Multi-Agent Simulations: Bay of Biscay Case Study," submitted to SimTechT, 2003.

Dobbie, James M. "Search For an Avoiding Target," *SIAM Journal of Applied Mathematics*, Vol. 28, No. 1: 72-86 (January 1975).

Eagle, James N. and Alan R. Washburn. "Cumulative Search-Evasion Games," *Naval Research Logistics*, Vol. 38: 495-510 (1991).

Ferber, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Harlow England: Addison-Wesley, 1999.

Hendler, James. "Making Sense out of Agents," *IEEE Intelligent Systems*, 32-37 (March/April 1999).

Horne, Gary E. “Beyond Point Estimates: Operational Synthesis and Data Farming,” in *Maneuver Warfare Science 2001*. Eds. Gary E. Horne and Mary Leonardi. Quantico VA: Marine Corps Combat Development Command, 2001.

Ilachinski, A. “Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Combat,” *Military Operations Research*, Vol. 5, No. 3: 29-46 (2000).

Jilson, Erik W. and Esref Mert. *Modeling Conventional Land Combat in a Multi-Agent System Using Generalization of the Different Combat Entities and Combat Operations*. MS thesis, Naval Postgraduate School, September 2001.

Law, Averill M. and W. David Kelton. *Simulation Modeling and Analysis, 3rd Ed*. Boston: McGraw-Hill, 2000.

Luce, R. Duncan and Howard Raiffa. *Games and Decisions: Introduction and Critical Survey*. New York: Wiley, 1957.

McCue, Brian. *U-Boats in the Bay of Biscay*. Washington DC: National Defense University, 1990.

Miller, John O. Class slides, OPER 671, Joint Combat Modeling I. School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, Summer Quarter 2002.

Parunak, H. Van Dyke and Sven Brueckner. “Entropy and Self-Organization in Multi-Agent Systems,” *Proceedings of the International Conference on Autonomous Agents*. 124-130. Montreal Canada, 2001.

Rapoport, Anatol. *Two-Person Game Theory*. Ann Arbor: University of Michigan Press, 1966.

Rocha, L. M. *From Artificial Life to Semiotic Agent Models: Review and Research Directions*. Los Alamos National Laboratory report, LA-UR-99-5475. Los Alamos NM, 1999.

Russell, Stuart and Peter Norvig. *Artificial Intelligence: A Modern Approach*. New York: Prentice-Hall, Inc., 1995.

Stewart, Theodor J. “A Two-Cell Model of Search For an Evading Target,” *European Journal of Operational Research*, Vol. 8: 369-378 (1981).

Sycara, Katia P. “Multiagent Systems,” *AI Magazine*, Vol. 19, No. 2: 79-92 (1998).

Tighe, T. R. *Strategic Effects of Airpower and Complex Adaptive Agents: An Initial Investigation*. Masters Thesis, AFIT/GOA/ENS/99M-09. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1999.

Waddington, C. H. *O.R. in World War 2*. London: Elek Science, 1973.

Washburn, A. R. "On a Search For a Moving Target," *Naval Research Logistics Quarterly*, Vol. 27: 315-322 (1979).

Vita

Captain Joseph C. Price graduated from TASIS England American School in Thorpe, Surrey, England. He entered undergraduate studies at the United States Air Force Academy in Colorado Springs, Colorado where he graduated with a Bachelor of Science degree in Aeronautical Engineering in May 1998. Upon graduation, he was commissioned into the Air Force.

His first assignment was to Davis-Monthan AFB, Arizona as a maintenance officer in August 1998, where he was assigned as the Avionics Flight Commander for the 355th Component Repair Squadron. In January 1999, he moved the 354th Fighter Squadron where he served as the Assistant Sortie Generation Flight Commander for the base's combat A-10 squadron. In July 1999, he moved to the 357th Fighter Squadron to become the Sortie Support Flight Commander for one of only two A-10 training squadrons in the Air Force. He remained with the 357th FS until January 2001, also serving as the Sortie Generation Flight Commander from March 2000 until he left. He was then assigned to the 355th Operations Support Squadron.

In August 2001, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to Headquarters, Air Force Material Command at Wright-Patterson AFB, Ohio.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 074-0188*
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**			
1. REPORT DATE *(DD-MM-YYYY)* 17-03-2003	**2. REPORT TYPE** **Master's Thesis**		**3. DATES COVERED** *(From – To)* Jun 2002 - Mar 2003
4. TITLE AND SUBTITLE GAME THEORY AND U-BOATS IN THE BAY OF BISCAY			**5a. CONTRACT NUMBER** ORCMORD02710145
			5b. GRANT NUMBER
			5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Price, Joseph C., Captain, USAF			**5d. PROJECT NUMBER**
			5e. TASK NUMBER
			5f. WORK UNIT NUMBER AFIT 2002-058
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765			**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GOR/ENS/03-18
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Modeling and Simulation Office ATTN: Dr. Numrich 1901 N. Beauregard St., Suite 500 Alexandria VA 22311			**10. SPONSOR/MONITOR'S ACRONYM(S)** DMSO
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.			
13. SUPPLEMENTARY NOTES			
14. ABSTRACT Analysis of a large combat campaign using game theory is difficult due to non-linearities and other soft factors which exist in a complex system. However, game theory can give decision makers insight into strategies and outcomes that can be utilized to maximize one's objective. Agent-based simulation provides the means to model complex systems with non-linearities, by allowing for interactions among independent "agents." This thesis investigates game-theoretic strategies in agent-based simulation, modeled after the Allied search for U-boats in the Bay of Biscay during World War II (WWII). It also looks into the effects of adaptation on strategies by comparison to fixed-strategy results.			
15. SUBJECT TERMS agent modeling, combat models, game theory, operational research			

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Raymond R. Hill, PhD, Wright State University
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	80	(937) 775-5150; email: Ray.Hill@wright.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18